http://www.likno.com

AllWebMenus
Professional
5
Copyright © Likno Software - All rights reserved - www.likno.com

User's Guide

AllWebMenus

The Complete Tool for Powerful and Effective Web Site Navigation

by Likno

Likno Software is proud to have developed AllWebMenus, the most versatile system available for your web site navigation needs. After years of rigorous development, AllWebMenus brings you a full set of features which make cross-browser menu-making easier for the professional and amateur web developer. Additionally, new browser versions are continuously supported, making AllWebMenus the most complete cross-browser menu-making tool available.

The present manual will assist all our users in taking the most out of AllWebMenus and create fantastic menus for their websites.

The possibilities are endless!

The Likno Software team

AllWebMenus User's Guide

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: 13-Jan-14

Special thanks to:

We would like to thank all of our users and customers that have contributed their comments and suggestions for improving AllWebMenus and the services we offer. It is our aim to listen to all of them carefully in order to bring the best possible result for them. We couldn't have done without you guys!

Table of Contents

Introduction

Part I

1 Introduction

1.1 What is AllWebMenus

Create Menus in 5 Easy Steps

AllWebMenus is a dhtml/javascript menu builder for web site navigation that requires absolutely no DHTML or Javascript experience from you. It creates cross-browser, popup or drop down menus that work alike in all browsers supporting DHTML (like Firefox, Internet Explorer, Google Chrome, Opera, Mozilla, Konqueror, Safari and more).

A javascript menu can be designed with a simple, tree-like approach. You can then fully customize it by using the Properties Pane or by choosing a predefined appearance from the Style Gallery. It can be either vertical or horizontal (popup or drop down menu), it can be movable, stay visible while scrolling, contain static or animated images, borders, colors and much more!

Once everything is set, you can compile and link your menu to your web pages in a fast and easy manner, without any code involved!

AllWebMenus will save you a significant amount of time and effort required to fully understand Dynamic HTML and Javascript code capable to support **all DHTML-enabled browsers!**

Major Features Include:

Compile and link menus to your web pages in no time and **without any code required**.

Supported Authoring Tools:
Adobe Dreamweaver with free Extension
FrontPage with free Add-In and Navigation Structure import into AllWebMenus
NetObjects Fusion with free component
Adobe Golive
Visual InterDev
HomeSite
HotDog
HTML Kit
1st Page 2000
CoffeeCup HTML Editor
Cool Page
CuteHTML
and many more...

Import FrontPage Navigation Structures into AllWebMenus.

Create Site Maps straight from the Menu Structure of menus.

Create menus fast by using pre-designed menu styles from the **Theme Gallery.**

Use **multiple menus** in the same page. (version PRO only)

Use many flexible ways to position your menus: Either using window coordinates or attaching it to any element inside the page.

Create menus for use inside **framed web sites**.

Use complete help for all properties and procedures. Press F1 while you are on a property to immediately receive **Quick Help** for that property.

Design both **horizontal and vertical menus** and use static or **animated images** for both the background and content.

Create effects by assigning different values to the item properties for three different item states: "**Normal**", "**Mouse Over**" and "**Mouse Click**" item states.

Position the menu anywhere on the screen and keep the menu visible in the same position while scrolling (**<Menu Floats on Scroll>** property).

Specify **Javascript Commands** to be executed when clicking on menu items.

Specify **Javascript Commands** to be executed when a **submenu appears/disappears** or even when the **Main Menu** appears/disappears.

Use both **vertical** and **horizontal submenus** at any possible combination.

Use **Titles** on submenus and allow the user to **drag the menu** from its Title.

Choose "**How**", "**When**" and "**Where**", **any** submenu **appears**.

Use "**Distance Between Items**" and assign a "**Status Bar Text**" for each item.

...and many more (view the complete properties listing for more info).

Short list of new features added on version 5:

(see updated, full list online)

- **Improved interface** and design with many functional additions

- **Slide Menus** (aka "sliding menus" or "slide-in menus") are now available.

- **Use your own HTML code** on selected menu items to achieve enhanced formatting or advanced functionality, from a simple working link, up to forms, flash objects, search boxes, even full-scale web pages!
(supported through the enhanced <Text> property)

- **Vertical Scrollbar** for vertical Groups now available
(supported through the enhanced <Type> property)

- **Multiple Rows and Columns** on submenus now available
(also supported through the enhanced <Type> property)

- The **Horizontal Menubar** feature now available for both Main Menu
and Submenu Groups along with several enhancements
(supported through the enhanced <Type> property)

- **New submenu appearance effects** added with the ability to set
custom appearance time delays
(supported through the enhanced “Submenu Group Appears” property)

- **New submenu disappearance effects** added with the ability to
set custom disappearance time delays
(supported through the new “Submenu Group Disappears” property)

- More stylish **dhtml menu Themes** are now included in the installation for free.

- **New Examples** of Javascript Menu Projects added.

- The <Positioning> property is completely re-designed and
enhanced with a new positioning option added; you can now
easily **position your dhtml menu relative to an image** existent in
your pages.

- Ability to **use custom variables instead of actual text** in
certain text-related properties has been added, allowing for
more “dynamic” content in javascript menus.

- **20 new Textures** added in Image Gallery.

- **20 new Buttons** added in Image Gallery.

- **54 new Bullets** and Arrows added in Image Gallery.

- **40 Icons** now available into the new "Icons" Category of Image
Gallery.

- The Apply Theme dialog now allows you to view and choose
Themes based on their <Type> ("Horizontal" or "Vertical" or "Vertical Sliding"),
facilitating your theme selection goals.

- **Complete HTML is now available on every item**, allowing you to
show anything you like inside the item, like flash objects or
even forms.
(supported through the enhanced <Text> property)

- **Different colors on each border side** (top, right, bottom, left)
now available
(supported through the enhanced <Border> property)

- The **Overline and Strikethrough** options are now available for
the item text.
(supported through the enhanced <Text Decoration> property)

- A "Refresh" button now appears at the top-right corner of any
property when its values are edited.

- The "Menu Positioning" shortcut has been added.

- The **"Manual Trigger Groups - Get ID and Trigger Code"** feature of the <Positioning> property is completely re-designed and enhanced, allowing for advanced positioning effects.

- The "Ignore script in Preview Pane" option has been added in the 'Project Properties -> Advanced Scripting' form.

- The **"Change Item to Hidden/Visible"** command has been added.

- **IE6 fix on select boxes** (combos) now comes with more options to cover all cases (https websites for example).

- The Main Menu Size now appears in Preview Pane, so that you have an estimation of how large the dhtml menu will render in actual browser.

- The "Add Item Before" action now available when building the Javascript Menu Structure.

…and many more!

Watch online demos on important features of AllWebMenus

1.2 What's new

Please check the new features, enhancements and fixes of AllWebMenus by visiting the following page online:
http://www.likno.com/whatsnew.html (PRO version)

Note: We do not maintain a separate "what's new" section for the LITE version, as changes related to the LITE-supported features are also included in the LITE version.

1.3 LITE vs. PRO - Where do they differ?

The LITE version is for the **home user** who wants to use **simple**, yet truly **cross-browser menus**, only for his or her personal web site.

The PRO version is for the **professional user** who wants to implement **feature-rich cross-browser menus**, for as many web sites as needed.

Both LITE and PRO versions use the **same optimized menu-building libraries**, allowing the LITE user to enjoy the same cross-browser support as the PRO user does.

Both AllWebMenus LITE and AllWebMenus PRO share the same engine of **extra optimized menu-building libraries**. You can select among many **menu styles** and **features** for your menus, with the **cross-browser**, **cross-platform** assurance carried by AllWebMenus.

The main differences between AllWebMenus LITE and AllWebMenus PRO are found in the user interface – which was **simplified** with the use of menu styles for LITE users – and some of the properties which are available for the menu implementation. The following is a comparison table that displays the differences in the features and properties supported by AllWebMenus and AllWebMenus PRO:

Features	LITE	PRO
Use of images	no	yes
Multiple menus	no	yes
Import FrontPage Navigation Structure	yes	yes
Create sitemaps	yes	yes
Custom compile folders	no	yes
Cross-Frame support	no	yes
Menu Optimization	yes	yes
FrontPage Support	yes	yes
Support for advanced positioning method 2	no	yes

* LITE version does not support different styles for different items, as the PRO version does. Many similar elements have a **common** style as specified in the Style Editor. In specific, there are four different style categories that define the appearance of the following menu elements:

- The Main Menu area
- The Items of the Main Menu (same style is applied to all Main Menu items)
- The Submenu areas (same style is applied to all Submenus)
- The Items of all submenus (same style is applied to all Submenu items)

To see the differences on all properties between LITE and PRO, please click on the following sections:

Group Properties

Item Properties

See also
Properties Overview

1.4 Why Register

The AllWebMenus unregistered version is fully functional in its menu design capabilities. It displays the designed menus on a **Preview Window** where you can test their appearance and behavior as if viewed on a browser.

Registering AllWebMenus actually lets you **compile and link your menus to web pages** and use them on the Internet. It also entitles you for **free customer support for 6 months** and **free updates for 1 year** after registering your AllWebMenus installation. The compiled menus are fully functional in all major browsers.

Apart from removing this restriction, there are many more good reasons to register your license:

- **Reliability**:
  You get more than 10 years of accumulated experience in our javascripts development.

- **Support**:
  You get quick and thorough answers from our dedicated support team plus detailed documentation.

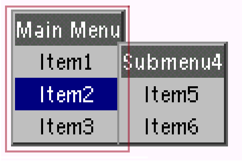
- **Value-added services**:
  You get free monitoring of your menus (optional) and get notified when their scripts are outdated due to enhancements in technology. Build your menus and worry no more!

- **Growth**:
  Your registration fee is invested in the rapid growth of AllWebMenus. More powerful features will be available to you.

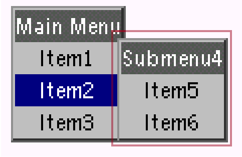
1.5 Terminology

Element
A generic name that describes any Main Menu, Submenu or Item

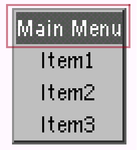
Main Menu
The first group of items forming the initial menu



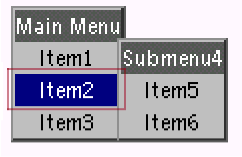
Submenu
Any group of items linked to a parent item



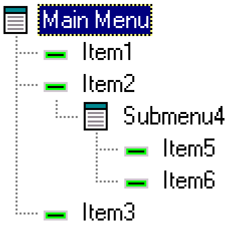
Header/Footer
Header is the top part of a Main Menu/Submenu. Bottom applies for the Footer.



Item
A selection that opens a submenu or a specified URL (or executes a Javascript command)

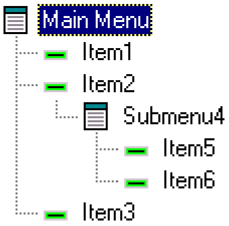


Child of element [A]
An element that directly derives from the element [A]



Ex: Submenu4 is a child of Item2
Ex: Item5 is a child of Submenu4
Ex: Item6 is a child of Submenu4

Parent of element [A]
The element from which the element [A] directly derives



Ex: Item2 is the parent of Submenu4
Ex: Submenu4 is the parent of Item5
Ex: Submenu4 is the parent of Item6

Theme Gallery
A collection of ready-made themes to apply to your menus.
See also: Apply Theme to menu

Style
The appearance and behavior properties of the element. Multiple items may use the same style.

Content
The specific text, image, etc. of the element.

Element Clipboard
Where the selected element is stored after a Cut or Copy.
See also: Copy Item/Submenu

The .awm, "menu definition" file:
This is the file type that AllWebMenus understands as a menu project. Let's say that you design a menu and you save it as *mainmenu.awm*. This file contains the definition of the menu elements along with its structure and can be opened by AllWebMenus for further development. This .awm file does not have to be placed in your web site's directory structure as it does not take part in the menu generation process (see below).

The .js, "compiled menu" file:
After finishing with your *mainmenu.awm* definition file, you need to compile it into a format understandable by the Javascript menu-building Libraries. Let's say that you compile it as *mainmenu.js* . This javascript file **has** to be placed in your web site's directory structure as it takes part in the menu generation process (usually the compiled menu is placed at root). Do **not** mistakenly compile it inside the C:\Program Files\AllWebMenus directory or any other working directory. The compiled menu is part of your uploaded WEB structure and therefore it should only be located there.

The "awmdata" folder:

This is the folder where the following elements are placed upon compilation:

- the *awmlibX.js* menu-building libraries and a *dot.gif* file (when the default "Javascript Menu" Genre is used) **or** just the *awmcsslib.js* menu-building library (when the "CSS Menu" Genre is used instead).

- the images used in the menu design (if any), located at a separate "Menu images" subfolder of the "awmdata" folder.

The dot.gif image file: (applies to the "Javascript menu" Genre only)
This is a **required** file used to implement the cross-browser rendering of the menu. It is located inside the *awmdata* directory along with the *awmlibX.js* libraries. If missing, a small line will appear on the left side of every menu item and the menu may not show properly in some browsers.
You also have the option to include the "dot.gif" image file inside the "Menu images folder". Read here when and why.
Implementations of the "CSS Menu" Genre do not need this file and do not produce it upon compilation.

The awmlibX.js, "Javascript menu-building Libraries":
(click here for definition)

The "menu-linking code":
(click here for definition)

1.6 About

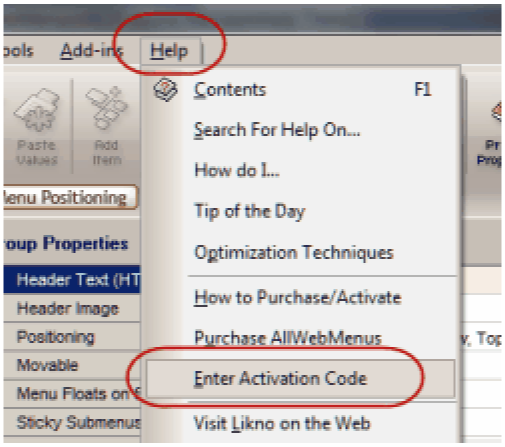
Likno Software is a software development company dedicated to creating quality software for demanding web authors. Our development team is 100% Internet-oriented, focused on creating tools that minimize your web development efforts.

For detailed **technical support** you can access the following URL:
http://www.likno.com/support_page.html

For **sales** questions you may contact us at **sales@likno.com**.

Our web site is located at http://www.likno.com

Note: In case you came here to **find your Installation ID**, it can be found at the **"Help -> Enter Activation Code"** command of the **application menu**, as shown below:


ools
Add-ins
Help
Contents F1
Search For Help On...
How do I...
Tip of the Day
Optimization Techniques
How to Purchase/Activate
Purchase AllWebMenus
Enter Activation Code
Visit Likno on the Web
Paste Values
Add Item
lenu Positioning
oup Properties
Header Text (HT
Header Image
Positioning
Movable
Menu Floats on
Sticky Submenus

Menus in 5 Easy Steps

Part II

2 Menus in 5 Easy Steps

2.1 Step 1: Creating your menu

Step 1

Creating your menu is quite easy:

- A new item can be added by right clicking with your mouse in the place where you want the Item and pressing the "Add Item" selection in the menu that is displayed.

- A new submenu group is added by right clicking the Item that will have the submenu and pressing on the "Add Submenu Group to Item" selection.

- By highlighting an item you can view on the right panel the full set of its properties in each of the three states of the menu (Normal, On Mouse Over and On Mouse Click). You can modify each of the properties by clicking on the appropriate property box and entering the value that suits your needs. This will allow you to fully customize the menu.

Watch online demos on important features of AllWebMenus

2.2 Step 2: Linking your menu to your web page

Step 2

The process of linking the produced menu to your web pages is remarkably simple:

The first thing you need to do is Compile the menu that you have created. This can be done by selecting Tools->Compile Menu. If you have not specified your "Site_Root folder" already, the form "Compile Properties->Folders & Files" will appear asking you to specify the folder where your web site is locally stored. This is where your menu will be compiled. The compilation creates the file and a set of library files that make up the full set of javascript scripts that will run when your menu is displayed.

After you compile the menu you will need to "Link" it to your web pages. You can do this by selecting Tools->Link Menu. The process of "linking" basically allows you to select the HTML documents that will carry the menu, and places the AllWebMenus Linking section at the beginning of those files. This linking section gives instructions to the web browser on the necessary files that will create your menu and points to the right directories for the images that you may have included in your menu.

Always make sure that the AllWebMenus menu-linking section is not inserted within table tags (<table><tr><td>) or <font> tags, as you may have problems with some browsers. The most safe position for the menu-linking code is at the beginning of the page, right after the <body> tag.

Note: AllWebMenus collaborates with all the common Web Authoring tools. In order to find out how your favorite Web Authoring tool integrates with AllWebMenus please **visit our special on-line section at** http://www.likno.com/web_authoring_tools.html

Frontpage Shared Borders users should check the "Support Shared Borders" checkbox (found in Tools -> Project Properties -> Compile Properties->FrontPage Support) and always link to the bottom shared border, irrespective of where the menu will appear on the page.

Watch online demos on important features of AllWebMenus

2.3 Step 3: Positioning your menu inside your web pages

Step 3

After linking your menu to your pages, you may easily position it at your preferred spot using any of the following methods:

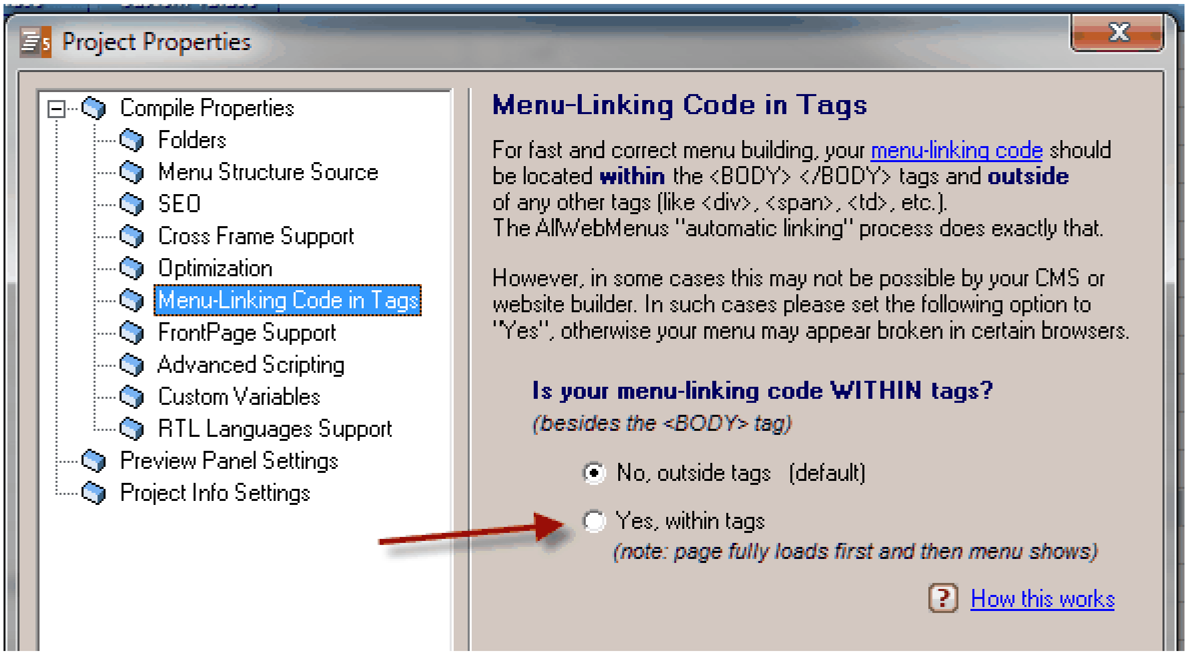
Window-Relative Positioning (default method)
Menu is positioned relative to a corner or edge of the browser window.

Element-Relative Positioning (positioning relative to a Page Element)
Menu is positioned next to a selected element of the page, after inserting a <SPAN> section at that particular position.

All methods are explained thoroughly in the "How Do I -> Position the menu relative to a page element" section.

Important Note!
Do not move the "AllWebMenus menu-linking code" from the beginning of the page! (i.e. from right after the <body> tag)
If you do so, you may have problems with some browsers!

If you cannot do otherwise, make sure that you check this option in "Project Properties":



Menu-Linking Code in Tags

For fast and correct menu building, your menu-linking code should be located **within** the <BODY> </BODY> tags and **outside** of any other tags (like <div>, <span>, <td>, etc.).
The AllWebMenus "automatic linking" process does exactly that.

However, in some cases this may not be possible by your CMS or website builder. In such cases please set the following option to "Yes", otherwise your menu may appear broken in certain browsers.

Is your menu-linking code WITHIN tags?
(besides the <BODY> tag)

● No, outside tags (default)
○ Yes, within tags
(note: page fully loads first and then menu shows)

How this works

(use this option only if you have to)

Positioning is implemented using the two positioning methods, **not** by moving this section from the beginning of the page. See Correct Placement of the menu-linking code for more info on why the position of the menu-linking code is irrelevant from the actual position where the menu is specified to appear.

Watch online demos on important features of AllWebMenus

2.4 Step 4: Uploading your menu

Step 4

Once you have compiled and linked your menu to your pages, you will need to upload the files to your web server. The files created by AllWebMenus during the compilation phase are:

./*menuname*.js	
./awmdata/	(folder)
./awmdata/dot.gif	*[image file, if "Javascript Menu"* Genre *is used]*
./awmdata/*.js	*[all javascript library files]*
./awmdata/menuname	*[folder with all image files used by the menu]*

where:

./	denotes the current directory
menuname	denotes the name you have chosen for the menu when you compiled

All those files have to be uploaded to your web server using your favorite FTP tool. Make sure that the directory structure is kept exactly the same as is displayed here and also pay careful attention in order to keep the case of the files when they are uploaded. You have also to make sure that the files are placed in the same location relative to your HTML files as they are locally, in order for the directories that were automatically set in the AllWebMenus linking section to point to the right structure on your web server.

Watch online demos on important features of AllWebMenus

2.5 Step 5: Saving your work

Step 5

Your menu can be saved in an awm project file so you can use/edit it in the future. Just use File->Save or File->Save As for this process.

Once your file is modified you will need to compile it again (but not link it) and upload the new *menuname*.js file to your web server. The *awmdata* folder needs to be re-uploaded only if you re-compile your menu with a newer AllWebMenus version (since it may contain updated menu-building libraries) OR you have added or altered images used by your menu.

Watch online demos on important features of AllWebMenus

User Interface

Part III

3 User Interface

3.1 Main window layout

The AllWebMenus Main window uses two tabbed sections:

The Menu Editor Tab

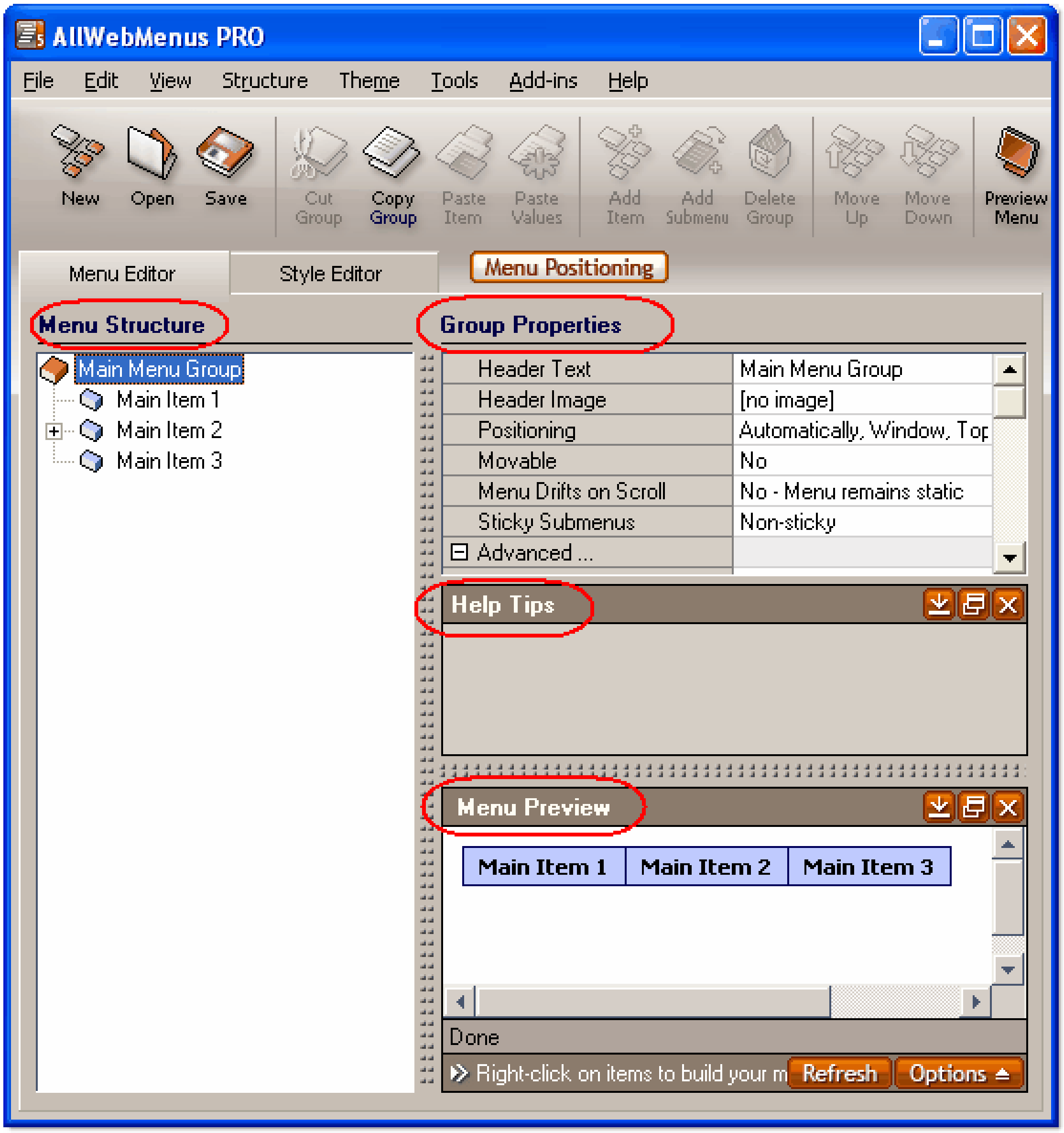
This section is split into four panes:

- **The Menu Structure Pane**
This is where the menu is structured by specifying its elements and its parent-child relationships.

- **The Properties Pane**
This is where the properties of each element are specified. The Pane changes its content according to the selected element (Group, Item)

- **The Help Tips Pane**
This is where a short description is displayed to provide quick help on the selected property.

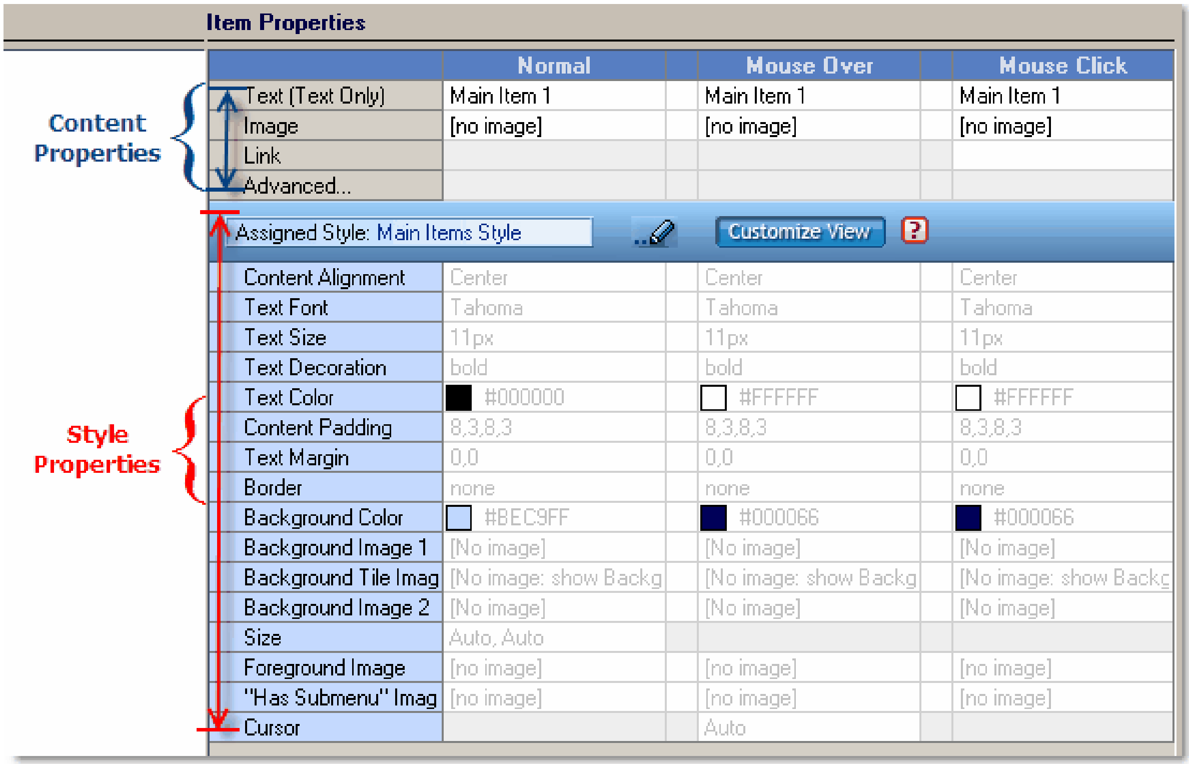
- **The Menu Preview Pane**
This is where your menu can be previewed in a WYSIWYG IE browser.



Header Text	Main Menu Group
Header Image	[no image]
Positioning	Automatically, Window, Top
Movable	No
Menu Drifts on Scroll	No - Menu remains static
Sticky Submenus	Non-sticky
⊟ Advanced ...	

The Properties Pane includes all available properties which are grouped into two categories.

The **Content Properties** and **Style Properties**:



	Normal	Mouse Over	Mouse Click
Text (Text Only)	Main Item 1	Main Item 1	Main Item 1
Image	[no image]	[no image]	[no image]
Link			
Advanced...			
Content Alignment	Center	Center	Center
Text Font	Tahoma	Tahoma	Tahoma
Text Size	11px	11px	11px
Text Decoration	bold	bold	bold
Text Color	■ #000000	□ #FFFFFF	□ #FFFFFF
Content Padding	8,3,8,3	8,3,8,3	8,3,8,3
Text Margin	0,0	0,0	0,0
Border	none	none	none
Background Color	□ #BEC9FF	■ #000066	■ #000066
Background Image 1	[No image]	[No image]	[No image]
Background Tile Imag	[No image: show Backg	[No image: show Backg	[No image: show Backg
Background Image 2	[No image]	[No image]	[No image]
Size	Auto, Auto		
Foreground Image	[no image]	[no image]	[no image]
"Has Submenu" Imag	[no image]	[no image]	[no image]
Cursor		Auto	

The Style Editor Tab

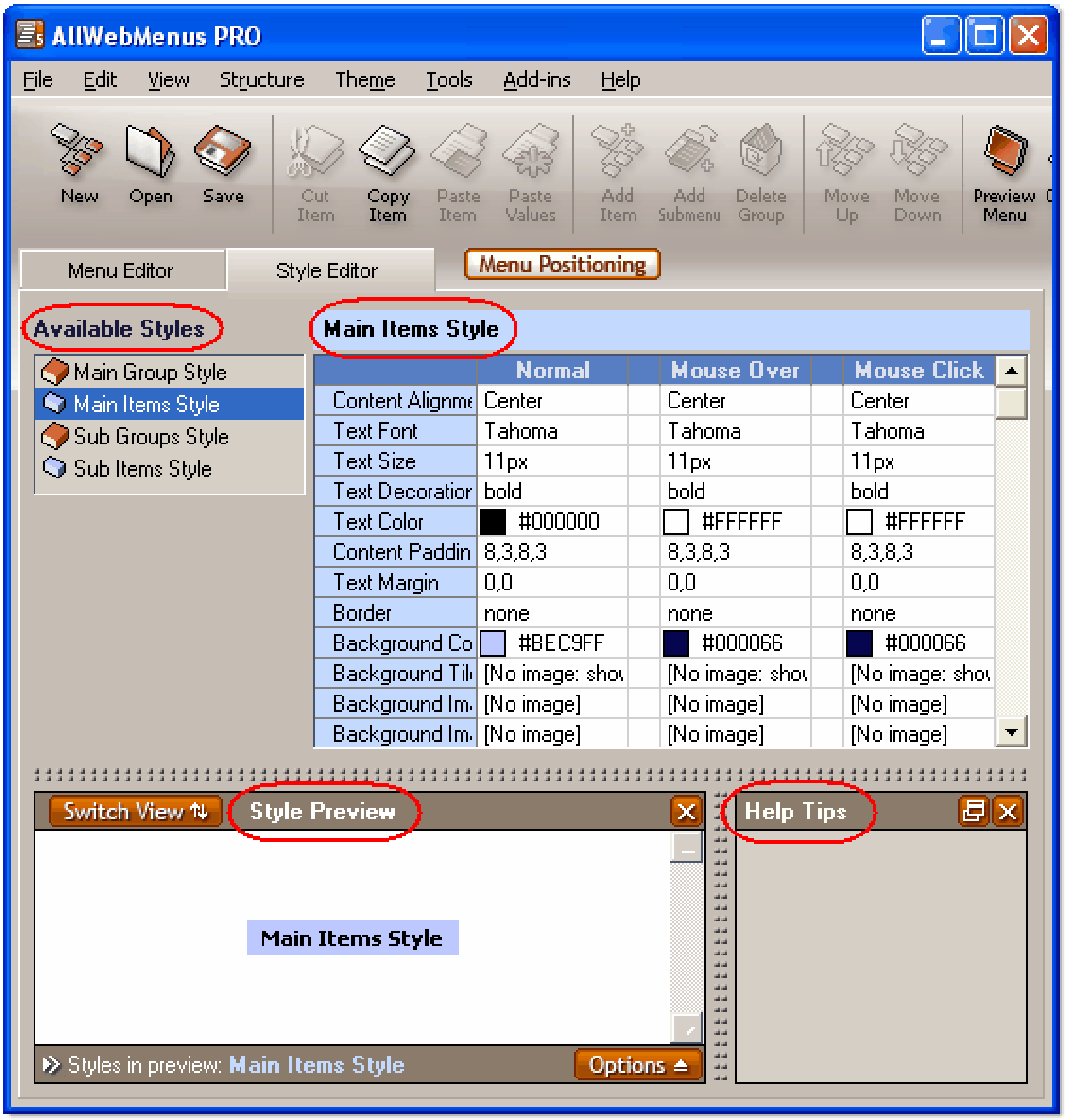
This section is split into four panes:

- **The Available Styles Pane**
This is where the available styles can be selected for further editing.

- **The Properties Pane**
This is where the properties of the selected style are edited. The Pane changes its content according to the type of the selected style (Group, Item).

- **The Help Tips Pane**
This is where a short description is displayed to provide quick help on the selected property.

- **The Menu/Style Preview Pane**
This is where your menu OR the selected style can be previewed in a WYSIWYG IE browser. If a group style is selected the preview pane displays the group along with the item that was previously selected.


AllWebMenus PRO
File Edit View Structure Theme Tools Add-ins Help
New Open Save Cut Item Copy Item Paste Item Paste Values Add Item Add Submenu Delete Group Move Up Move Down Preview Menu
Menu Editor
Style Editor
Menu Positioning
Available Styles
Main Group Style
Main Items Style
Sub Groups Style
Sub Items Style
Main Items Style
Switch View
Style Preview
Help Tips
Main Items Style
Styles in preview: Main Items Style
Options

	Normal	Mouse Over	Mouse Click
Content Alignme	Center	Center	Center
Text Font	Tahoma	Tahoma	Tahoma
Text Size	11px	11px	11px
Text Decoratior	bold	bold	bold
Text Color	■ #000000	□ #FFFFFF	□ #FFFFFF
Content Paddin	8,3,8,3	8,3,8,3	8,3,8,3
Text Margin	0,0	0,0	0,0
Border	none	none	none
Background Co	□ #BEC9FF	■ #000066	■ #000066
Background Til	[No image: shov	[No image: shov	[No image: shov
Background Im	[No image]	[No image]	[No image]
Background Im	[No image]	[No image]	[No image]

3.2 Keyboard shortcuts

There are several shortcuts defined for quicker access to frequently used functions:

- **F1**
Provides context sensitive help for each property

- **F12**
Opens the Preview Menu window

- **Ctrl + F12**
Opens the Compile Menu window

- **Shift + Ctrl + F12**

Opens the "Link Compiled Menu to Web Pages" window

- **Ctrl + Z**
Undo

- **Ctrl + Y**
Redo

- **Ctrl + N**
Creates a new AllWebMenus structure

- **Ctrl + O**
Opens an .awm menu project

- **Ctrl + S**
Saves the current AllWebMenus structure

- **Ctrl + X**
Cuts the selected element

- **Ctrl + C**
Copies the selected element

- **Ctrl + V**
Pastes the selected element

- **Ctrl + A**
Pastes the values of the selected element only

Commands

Part IV

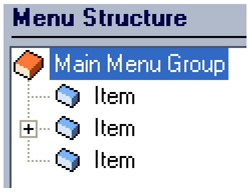
4 Commands

4.1 Command: File

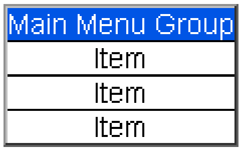
New

Creates a new AllWebMenus menu structure.
The new menu consists of three items and all properties are set to the default style:

This is how a new menu structure looks:



This is how a new menu design looks:



Open

Opens an AllWebMenus project.
The AllWebMenus files are found under the **.awm** extension.

Save

Saves the current menu project into an ***.awm** file.

Save As

Saves the current menu structure with a different filename.

Open Example Menu Project

Opens a list of example projects to select and load to the application.

Export Project

Exports the loaded project in order to move it to other PCs without having problems with the file paths of the linked images.

Exporting the project adds all linked images into a single folder, called "*menuname*-assets", which is located next to the exported .awm file.

Note that this action does not affect the file paths of the loaded .awm project. Your loaded .awm project continues to use the existing image paths.

Import Project (PRO only)

Imports the menu structure from another project (.awm) into the current project.

Exit

Closes the AllWebMenus application.

4.2 Command: Edit

Undo

Lets you undo the last action you made.

Redo

Lets you "cancel" an Undo action.

Cut

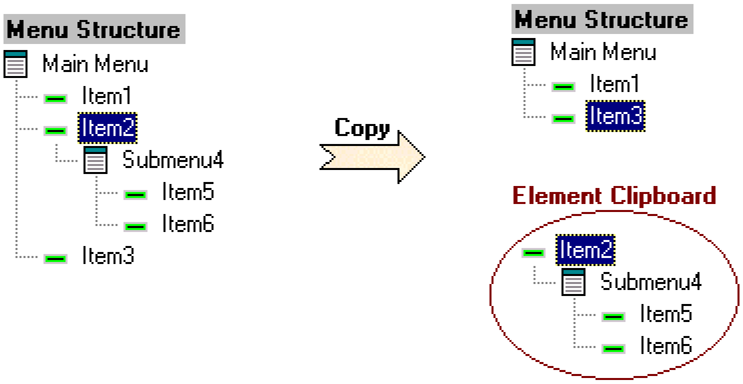
Copies the selected element (Group or Item) to the Element Clipboard and then deletes it from the menu structure.

Copy

Copies the selected element to the Element Clipboard.

If the selected element has any children (i.e. elements linked to it) the **copy** will also apply to them. This means that the element clipboard will contain both the element **and** its child elements too.

See an example:

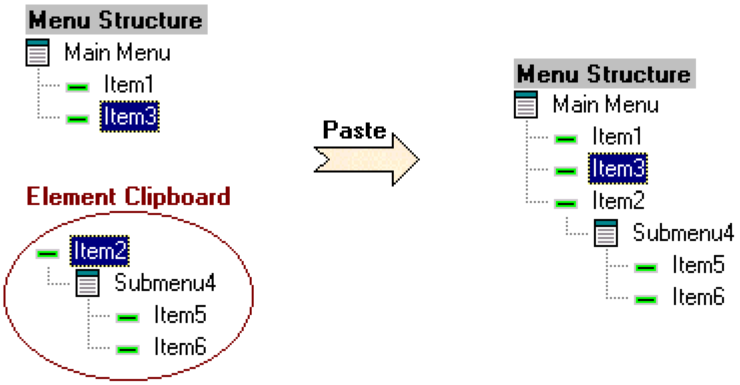


Paste

Inserts the content of the Element Clipboard into the menu structure.

When pasting the clipboard element, a new element is added in the menu structure along with its children.

See an example:



Since groups are always linked to items, a group will not be pasted when the selected element is also a group.

Paste Values

Lets you paste the values of the copied element to the selected element (does not create a new element in the menu structure). You can always select a subset of the copied values to paste to your target elements.

Note that you can only paste to elements that are similar to the copied element; i.e. if you copied an item you can only paste its values to other items, not to groups.

4.3 Command: View

This command lets you hide or show certain panels that may be useful but not required for the use of the program.

While we do recommend that you keep these panels visible you may hide them if your monitor space is limited.

The panels that can be hidden are the following:

- Help Tips panel
- Menu Editor Preview panel
- Style Editor Preview panel

4.4 Structure

4.4.1 Command: Add Item - Before Selected

Adds a new item in the menu structure **before** the selected item.

The new item can always be placed somewhere else by using the **Move Item Up/Down/Left/Right** or the **Cut/Paste** functions.

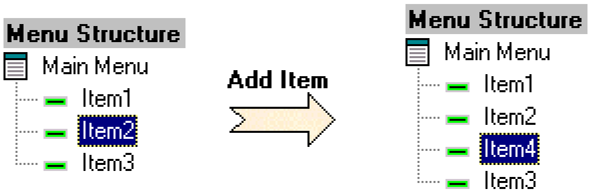
See also
Add Item - After Selected

Delete Item

4.4.2 Command: Add Item - After Selected

Adds a new item in the menu structure **after** the selected item.

If the selected element is an item, the new item is placed below it.



The new item can always be placed somewhere else by using the **Move Item Up/Down** or the **Cut/Paste** functions.

See also
Add Item - Before Selected
Delete Item

4.4.3 Command: Add Child Item (in Submenu Group)

Creates a submenu group for the selected item and adds a child item to it.

Since submenu groups are always linked to items, your selection should be an item to perform this operation.

You can always transfer the new submenu group to another parent item by using the Cut & Paste functions.

See also
Delete Submenu

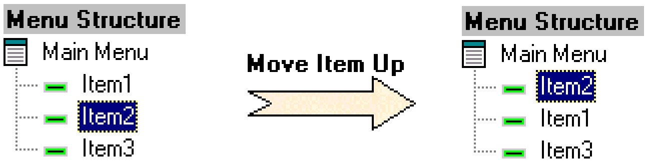
4.4.4 Command: Add Separator

Adds a separator on the selected item.

For more information on how this works read the <Item Separator> topic.

4.4.5 Command: Move Item Up

Moves the selected item up by one position.

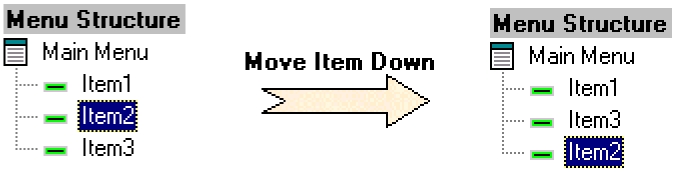


This action will not move the item outside its parent group. This can be easily done by using the "Move Item Left/Right" OR the "Cut and Paste" functions instead.

See also
Move Item Down
Move Item Left
Move Item Right

4.4.6 Command: Move Item Down

Moves the selected item down by one position.



This action will not move the item outside its parent group. This can be easily done by using the Cut and Paste functions instead.

See also
Move Item Up
Move Item Left
Move Item Right

4.4.7 Command: Move Item Left

Moves the selected item **one level up**, as sibling of the parent item.

This action moves the item **outside** its parent group.

See also
Move Item Right
Move Item Up
Move Item Down

4.4.8 Command: Move Item Right

Moves the selected item **one level down**, as child of the previous item.

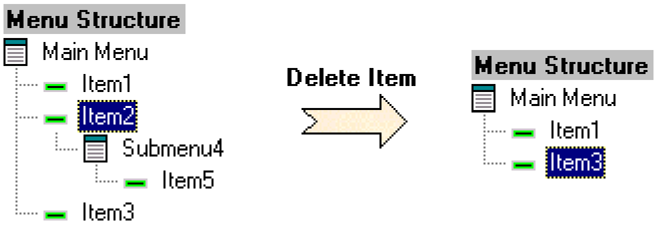
This action moves the item **outside** its parent group, into a new submenu group of the parent.

See also
Move Item Left
Move Item Up
Move Item Down

4.4.9 Command: Delete Item

Deletes the selected item from the menu structure.

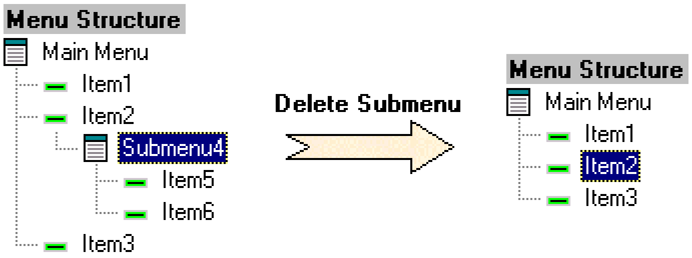
If the selected item has any child elements, the child elements are deleted as well.



See also
Add Item

4.4.10 Command: Delete Entire Submenu Group

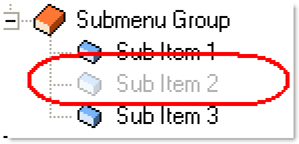
Deletes the selected submenu group along with all its child elements.



4.4.11 Command: Change Item/Group to Hidden/Visible

Toggles the status of an Item or Group from Visible to Hidden and vice versa.

A hidden item or group does not show up in the compiled menu and it is shown in the Menu Structure as grey:



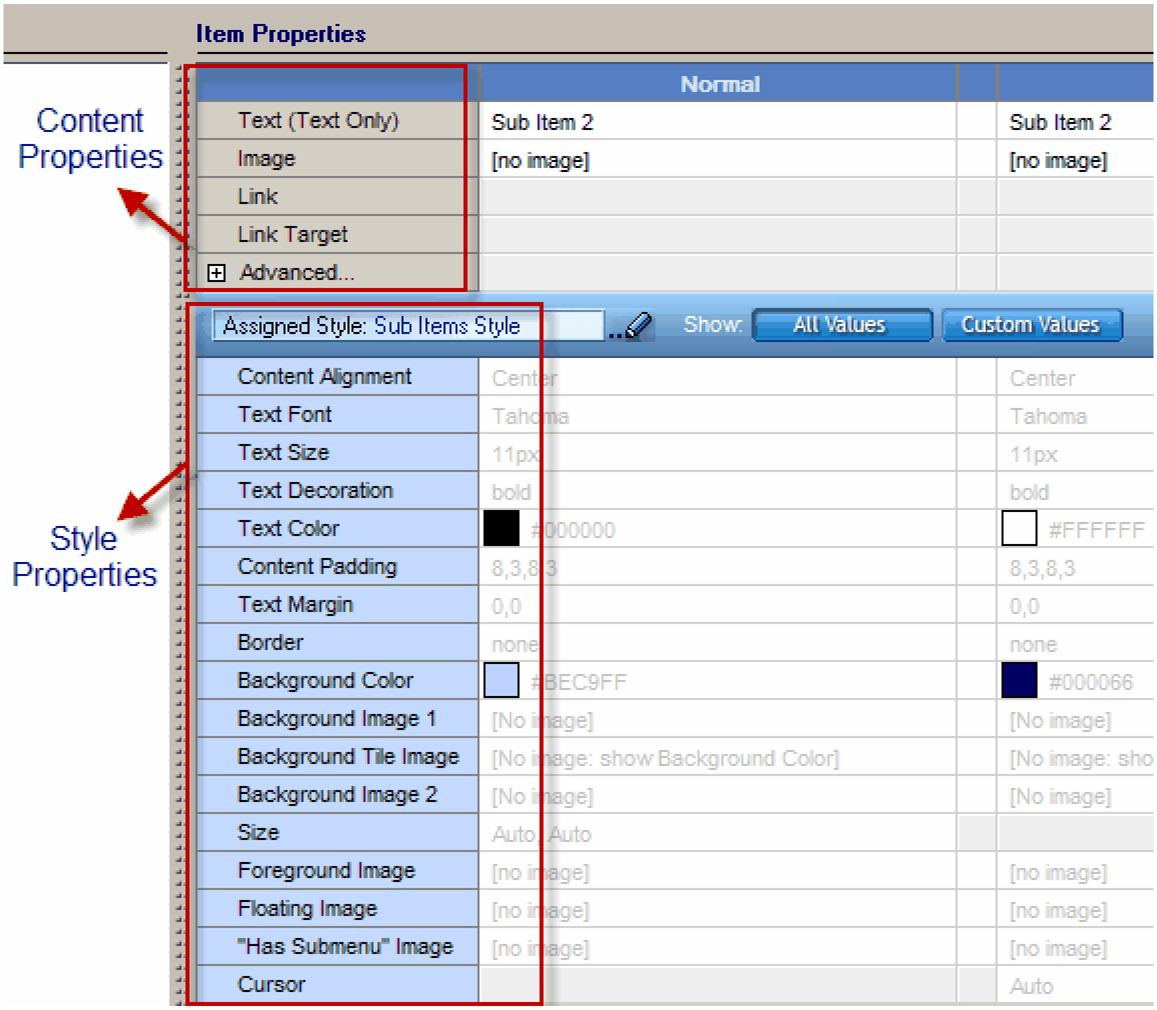
4.5 Theme

4.5.1 Command: Apply Theme to Menu

Opens the **Theme Gallery** to choose from a collection of predefined themes and apply it to your menu.

This action replaces all styles used in your menu with the styles contained in the theme.

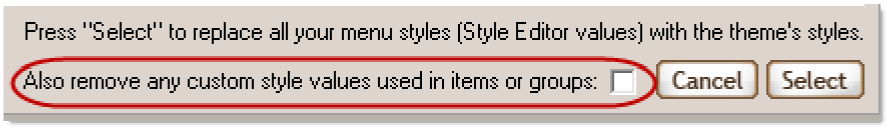
Therefore, your menu appearance changes while its content remains intact ("Content Properties"):



Item Properties

	Normal		
Text (Text Only)	Sub Item 2		Sub Item 2
Image	[no image]		[no image]
Link			
Link Target			
⊞ Advanced...			
Content Alignment	Center		Center
Text Font	Tahoma		Tahoma
Text Size	11px		11px
Text Decoration	bold		bold
Text Color	■ #000000		□ #FFFFFF
Content Padding	8,3,8,3		8,3,8,3
Text Margin	0,0		0,0
Border	none		none
Background Color	□ #BEC9FF		■ #000066
Background Image 1	[No image]		[No image]
Background Tile Image	[No image: show Background Color]		[No image: sho
Background Image 2	[No image]		[No image]
Size	Auto, Auto		
Foreground Image	[no image]		[no image]
Floating Image	[no image]		[no image]
"Has Submenu" Image	[no image]		[no image]
Cursor			Auto

When choosing a theme you also have the following option: (version PRO only)

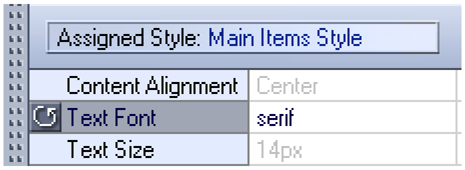


Press "Select" to replace all your menu styles (Style Editor values) with the theme's styles.

Also remove any custom style values used in items or groups: □

This option let's you decide whether you want the custom style values to be removed (and be replaced by the style properties of the applied theme) or keep these style values intact. The default behavior for these customizations (if any) is to be removed.

As an example of style customization, see the <Text Font> property below which has been customized for a particular item:



Assigned Style: Main Items Style

Content Alignment	Center
Text Font	serif
Text Size	14px

To read about **opening an EXAMPLE project** (2nd tab) click here.

See also
Export New Theme
Command: Open Example Menu Project

4.5.2 Command: Save Custom Theme

Creates a new theme (custom) based on the appearance of your Available Styles found in the Style Editor.

- The theme's Main Menu Group will get the "Main Group Style" appearance and its items the "Main Items Style" appearance.
- The 2nd level groups (first submenus) of the theme will get the "Sub Groups Style" appearance and their items the "Sub Items Style" appearance.
- The 3nd level groups (further submenus) of the theme will get the "Sub Groups Style+" appearance and their items the "Sub Items Style+" appearance.

You can also view and apply your new theme to existing menus by using the Apply Theme to Menu command. Custom themes are shown with an asterisk at their end (*).

For more information on how to create your **own custom themes**, view instructions online.

See also
Apply Theme to Menu

4.5.3 Command: Export all Custom Themes

Custom themes (the ones you save) can be exported, so that you keep them as a backup or import them to another AllWebMenus installation.

Custom themes are the ones you create (not provided by Likno Software) and have an asterisk at their end (*).

See also
Apply Theme to Menu

4.5.4 Command: Import to Custom Themes

Custom themes (the ones you have saved and exported before) can be imported to your AllWebMenus installation.

Custom themes are the ones you create (not provided by Likno Software) and have an asterisk at their end (*).

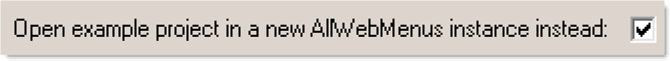
See also
Apply Theme to Menu

4.5.5 Command: Open Example Menu Project

Opens a list of example projects to select and load to the application.

Note that the Example Menu Project is not a Theme, which you can apply to your loaded menu. This means that your loaded project will have to close and the example project will open instead. You can then start building a new menu based on the example.

If you want to view the properties of the example project without closing your loaded project, you can check the following option:



Open example project in a new AllWebMenus instance instead: ☑

To read about applying a THEME to your menu (1st tab) click on the link below.

See also
Command: Apply Theme to Menu
Command: File

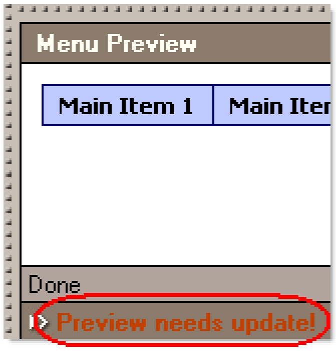
4.6 Tools

4.6.1 Command: Refresh Menu Preview

Displays the menu design on a WYSIWYG IE browser pane.

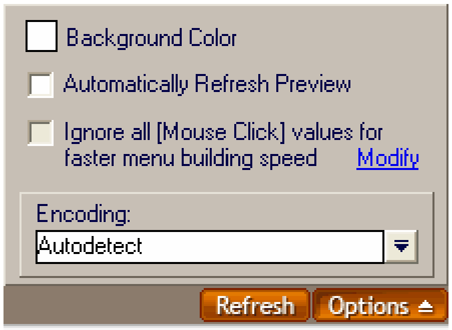
This is a fast way to view the changes made during the menu's design phase. It is a **WYSIWYG** window that will help you experiment with your menu properties, observe its behavior and implement its design exactly the way you like.

If the preview window is open and a change is made at the properties or in the menu structure, the preview window signals its non-updated status by the bottom message colored in red.



To show the latest changes in your menu design, press F12 or click the Preview Menu icon in the application toolbar or the Refresh button in the Menu Preview window. The preview will then be updated.

Note that you can have your preview automatically updated every time you make a change to the menu by checking the related check box at the options below:



□ Background Color

□ Automatically Refresh Preview

□ Ignore all [Mouse Click] values for faster menu building speed Modify

Encoding: Autodetect

Just have in mind that this is not recommended for very large menus as the automatic refresh may slow down the application response to your actions.

You also have a shortcut to ignore the Style values of the [Mouse Click] state for increasing your menu-building speed.
Please read the Optimization Techniques for more info on this option.

Note on Item Text property and Preview: If you use "HTML Formatting" on an item and you refer to local images or other assets there (eg: *src="image1.png"*) the Preview Pane cannot show them (but actual website will, if in proper relative position). Still, there is a way for the Preview Pane to also show them: you can add these assets into a "*projectname*-assets" folder, located in the same folder where the project is located (eg: *"menu-assets"*).

4.6.2 Command: Compile Menu

Compiles the .awm file into a .js file that has to be linked to your web pages.

To use a menu in your web pages you must take the following four steps:
Step 1. **Compile** the menu (note: spaces are not allowed in filename)

Step 2. **Link** the compiled menu to your web pages using the related AllWebMenus command
Step 3. **Upload** the necessary menu files and folders (see a description of these elements below)
Step 4. **Upload** the web pages that contain the menu link
(note: steps 2 & 4 are not necessary if you have already linked the menu to your web pages and you only want to update your menu. They may be necessary if you are using a new AllWebMenus version with improved menu-linking code)

How it works:

When compiling a menu as ***menu*.js**, the following elements are created inside the same directory:

1. The core **menu.js** Javascript file.

2. The **awmdata** directory (or your custom directory as defined in Project Properties > Compile Properties > Folders & Files > Advanced).
 This directory contains the necessary Javascript menu-building Libraries (awmlibX.js or awmcsslib.js) along with any images used by the compiled menu.

The **awmdata** directory (and its contents) is **required** and must always reside in the same directory with the ***menu*.js** file.
Therefore, it is critical that **you do not forget to upload the awmdata directory along with the *menu*.js file!**

Example
Let's say that the local root folder of your site is the **c:\mysite** folder and that the contents of your web site are:

| --- |
| c:\mysite\index.html |
| c:\mysite\companyinfo.html |
| ... |
c:\mysite\faq.html

You now design a menu in AllWebMenus, called **mysitemenu.awm**, and you compile it into a file called **c:\mysite\mymenu.js**.

Your local web site now contains:

| -- |
| c:\mysite\index.html |
| c:\mysite\companyinfo.html |
| ... |
| c:\mysite\faq.html |
| **c:\mysite\mymenu.js** |
| **c:\mysite\awmdata\awmlibX.js** |
| **c:\mysite\awmdata\dot.gif** |
| **c:\mysite\awmdata\mymenu\image1.gif** |
| **...** |
c:\mysite\awmdata\mymenu\image8.jpg

The elements in **bold** should be **uploaded** on the server in the same directory structure where they are locally created.

If you don't upload them or you misplace them on the server, the menu will produce a run-time error and will fail to appear due to the following reasons:

- **misplacement of the mymenu.js file**: all web pages linked to the compiled menu will not be able to find it.
- **misplacement of the awmdata directory**: the mymenu.js file will not be able to call the necessary awmdata*.js javascript libraries that build the menu.

Notes:

The "**mymenu**" name was chosen as an example. If you name the compiled menu "menu1", then you should find and upload a menu1.js file and an awmdata folder.

You can compile the menu anywhere you like **within** your site's directory structure. This decision should not be affected by the location of the pages you want to link the menu to.
For example, if you want to link the menu into a Dreamweaver template, you are **not** required to compile the menu into the Templates directory, as you will have to upload the Templates directory on the server too.
In general, simply compile your menu into your **root** directory or into a manually-created subdirectory and you will be able to link it to any page, shared border, template or library of your web site directory structure.

Note for "CSS Menu" Genre implementations:
The *awmlibX.js* and *dot.gif* files are produced when the default "Javascript Menu" Genre is used. In "CSS Menu" Genre implementations, the ***awmcsslib.js*** menu-building library is produced instead.

Notes for FrontPage users:
1. To properly use AllWebMenus with Shared Borders please read carefully the "Compile Properties, FrontPage Support" section.

2. The menu should not be compiled inside the *_borders* folder or any other special folders used by the authoring tool to manipulate common objects. The obvious reason is that these folders are not uploaded on the server when publishing the site. Instead, compile the menu in **root** or in any other manually-created subdirectory.

See also
Link Compiled Menu to Web Page(s)
Compile Properties: Folders & Files
Compile Properties: Cross-Frame Support
Compile Properties: Optimization
Compile Properties: FrontPage Support
How Do I Use my designed menu on my web pages

4.6.3 Command: Project Properties

Allows you to specify the following properties and options for your loaded menu project:

Preview Panel Settings

Project Info Settings

Compile Properties

Folders & Files

Cross-Frame Support

Optimization

FrontPage Support

Advanced Scripting

Custom Variables

RTL Languages Support

4.6.3.1 Compile Properties

To read more info on a "Compile Properties" topic, please click on its respective link below:

Folders & Files

Menu Structure Source

SEO

Cross-Frame Support

Optimization

Menu-Linking Code in Tags

FrontPage Support

Advanced Scripting

Custom Variables

RTL Languages Support

Menu Printing

4.6.3.1.1 Compile Properties: Folders

Here you specify the output folders and files to be created when compiling your menu project.

The **Site_Root folder** is the location where your web site (your main index page) is locally stored. It is also referred as the "Site Root" folder in other applications. It is very important that you specify the correct folder here, as it is used in both the compile process and in the <Link> item property for browsing the site pages.

The **Compiled menu name** is the name of the javascript file that is generated when you compile the menu. It defaults to *menu.js* but you may as well change it to something else. Note that you can use multiple menus in the same website by having different projects compiling menus under different "Compiled menu name" properties.

Once you setup your "Site_Root folder", you are able to edit the "**Compiled subfolders**" which include the following: (version PRO only)

Menu file subfolder

This specifies a custom subfolder in your web site where your menu will be compiled. The default folder where the menu gets compiled is the Site Root of your web site. Note that the folder needs to exist prior to selecting it.

When compiling a menu a default **awmdata** folder is also created under the same directory. This folder contains the necessary Javascript menu-building Libraries along with any images used by the compiled menu. Instead of using the **default** awmdata folder, you may use **two separate custom folders** for the following entities:

Javascript libraries subfolder

awmlibX.js The **required** Javascript menu-building Libraries. These are the javascript files containing all the functions that build the menu and drive its behavior.

dot.gif A **required** image file used to implement the cross-browser rendering of the menu in the default "Javascript Menu" Genre implementations. Read more below.

There is no apparent reason to change the menu-building libraries folder, so we recommend that you keep the default **awmdata** value.

Menu images subfolder

.jpg, or .gif files(all images used by the menu, except of the "dot.gif" special image that resides in the "Javascript libraries" folder. If not uploaded, the menu will still display all items but without images. The missing image sign ("**x**") may also appear in place of those images)

Custom folders may be required when two or more menus are used in the same web site and images from both menus use the same names. This is the usual situation that may require you to have a different subfolder for the produced images (default behavior).

Custom folders may contain **relative** paths too (eg: ../). If you do so, the folder location will be relative to the location of the compiled **menuname.js** file.

Examples:

Libraries Folder:	awmdata
Images Folder:	awmdata/menu
Libraries Folder:	allwebmenusdata
Images Folder:	allwebmenusdata/menu
Libraries Folder:	../allwebmenusdata
Images Folder:	../allwebmenusdata/menu
Libraries Folder:	allwebmenuslibs
Images Folder:	allwebmenusimages

Libraries Folder:	../allwebmenuslibs
Images Folder:	../allwebmenusimages

The 'Place "dot.gif" in' option

The file "dot.gif" is a **required** image file used to implement the cross-browser rendering of the menu. If missing, a small line will appear on the left side of every menu item and the menu may not show properly in some browsers.

Although it is an image file, it is inserted inside the "Javascript libraries" folder (eg: "awmdata") by default. This is absolutely fine for the vast majority of server settings.

However, in some rare server settings you may need to have it inserted into the "Menu images" subfolder by using the related **'Place "dot.gif" in'** option. In that way all images are placed together and only javascripts remain inside the "awmdata" folder.

On what rare cases is this option useful?
See below an email sent by a customer:

> Hello,
>
> I realize that after compiling a menu, the **dot.gif** file is always
> placed in the same folder as the library files. Unfortunately, our web
> host demands separation of image files (e.g., jpeg, gif) from
> executable files (e.g., html, js, etc.) in different folders.
>
> What is the best way to accomplish this?

This is exactly the case where the customer should change the option **from default to "Menu images folder".**

Note: *The dot.gif is used on "Javascript Menu" Genre implementations only. Implementations of the "CSS Menu" Genre do not need this file and do not produce it upon compilation, so this option has no effect when this Genre is used.*

4.6.3.1.2 Compile Properties: Menu Structure Source

Here you specify how your menu is populated with menu items (and submenus).

Option 1: using the project's structure, i.e. items that you add to your AllWebMenus project. These items are fetched from its compiled project's file **(default option)**,

OR

Option 2: "on-the-fly", using <UL><LI> items fetched from the loaded page that shows the menu.

Option 1: Populate menu inside this project, with items specified in Menu Structure (default)

With this option your menu uses the structure you create and edit within the AllWebMenus application.

When you compile, this structure gets embedded to a single javascript file and all pages linking to this file use that menu structure.

Contrary to the <UL><LI> option:

- You do not have to add menu items (<UL><LI> HTML code) on every single page of your website. When you make a change to the structure, you just re-compile and re-upload the related files to your server and you are done.

- The menu appearance can also use *custom*, item-specific style values (besides the styles of Style Editor).

- The menu appears faster (gets built while the page is still loading, not after).

- You can achieve SEO by either linking a sitemap HTML file to your home page (or to any other page indexed by search engines), OR still adding the <UL><LI> code to your pages (from option 2).

If you use the UL/LI code in your pages, do not get confused and think that the menu will ignore the "within project" items and populate itself using that UL/LI code. As long as you have checked the "Populate menu inside this project" option (1st option), the <UL><LI> code serves only for SEO purposes; not for populating the menu.

Option 2: Populate menu inside your page, through UL/LI items

With this option your menu *ignores* the project's menu structure (which is located in a javascript file) and rather builds its structure "on-the-fly", based on <UL><LI> elements added (by you) in your page's HTML code.

This method has the following characteristics:

- Search engines can crawl your menu structure on any page where your menu is shown. However, this can still be achieved through a single sitemap file (see how), so this should not necessarily be the reason to use that option.

- You can use a different menu structure on different pages. Since the menu is built "on-the-fly", based on <UL><LI> tags inside your pages' HTML code, you can have different tags on different pages.

- The menu appearance is based on the standard styles of the project's Style Editor only. Any *custom*, item-specific style values are *ignored*.

- The project's structure is still used in "Menu Preview Pane" in order to show you sample items when designing your menu (but it is not used on your actual pages, as this is substituted by your <UL><LI> items found there).

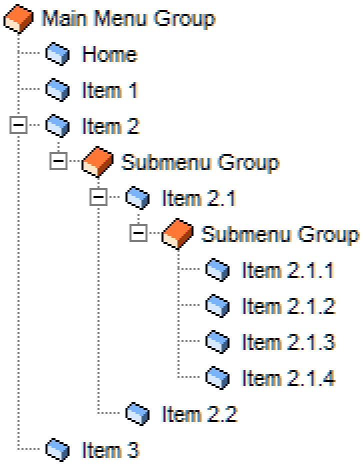
- The menu always appears after the page is fully loaded.

Example:

```
<ul id="awmULLI-menu" style="display:none;">
  <li><a href="index.html">Home</a></li>
  <li><a href="item1.html">Item 1</a></li>
  <li><a href="item2.html">Item 2</a>
    <ul>
```

```
      <li>Item 2.1
        <ul>
          <li><a href="item_211.html">Item 2.1.1</a></li>
          <li><a href="item_212.html">Item 2.1.2</a></li>
          <li><a href="item_213.html">Item 2.1.3</a></li>
          <li><a href="item_214.html">Item 2.1.4</a></li>
        </ul>
      </li>
      <li><a href="item_22.html">Item 2.2</a>
      </li>
    </ul>
  </li>
  <li><a href="item3.html">Item 3</a>
  </li>
</ul>
```

The above <UL><LI> HTML structure represents the following menu structure:



Notes:

- If you cannot use the **"id"** attribute in your HTML (in "WordPress" for example), you can use the **"class"** attribute instead:
`<ul class="awmULLI-menu" style="display:none;">`

- The **"id"** attribute is not required if you choose the **"1st UL/LI structure found in page"** population option and make sure that the UL/LI structure that specifies the menu items is before any other UL/LI structure in the page (or it is just the only one).

<u>How the <UL><LI> population method works:</u>

The AllWebMenus engine parses the <UL> structure that uses the same ID as the one specified within the AllWebMenus project (usually this defaults to "awmULLI-menu", unless you compile your menu with a different name or you change it manually).

Each <UL> represents a submenu (group), each <LI> represents an item.

The <A> tag contains the URL that the item opens when clicked.

The appearance of all items is specified within AllWebMenus, **not** through your website's CSS. In particular, the styles in Style Editor are used to create your menu theme (appearance/behavior).

This is why the <A> tags are not shown as underlined (for example) or using your website's CSS definition.

Your items are formatted based on the **"HTML - Formatting"** mode (see different modes), therefore you can still make some changes in appearance through HTML. The last Main Menu item in the sample <UL><LI> code shows this (it uses an <i> tag on its last 2 words). Note that in this mode, the mouse handling of the HTML tags you use within an <LI> tag is ignored (it is passed to the AllWebMenus engine instead).

Finally, you can also utilize the **"HTML - Full"** mode (see different modes) if you want to have full control even for the mouse events (note: this may not work for all HTML combinations!). The last item of the "Item 2" submenu in the sample <UL><LI> code shows this:

```
<li>Manual <span style="font-weight:bold;
color:#F00;">styling.</span>. Here <b>is a</b> <a
href="item_26.html">link</a>. <br>Item is rendered in &quot;Full
HTML&quot; mode</li>
```

If you use the above code in a menu, you will see that the <A> tag renders the "link" text based on your website's CSS (underlined for example) and is clickable, while the rest of the item area is not clickable.

AllWebMenus decides automatically if the "HTML - Full" mode applies or not, based on the following rule:

Is the <A> tag the **only** content within the <LI>?

YES *example:* `<li><a href="index.html">Home</a></li>`
=> the item is rendered based on the "HTML - Formatting" mode
(tags are used only for formatting, not for actions)

NO *example:* `<li>This is a <a href="page.html">link</a>.</li>`
=> the item is rendered based on the "HTML - Full" mode
(tags are used both for CSS-based formatting AND actions)

IMPORTANT NOTE when triggering the "HTML - Full" mode!
Not all HTML tags can work in combination with the engine's own styling, so avoid using complex customizations through the "HTML - Full" mode. Support cannot be given for problems when using this mode, as the <UL><LI> feature is strongly intended for use with simple items (like `<li><a href="index.html">Home</a></li>`).

About the *"display:none;"* style attribute:

The AllWebMenus engine automatically hides the <UL><LI> structure when it detects it, so that these links do not visually interfere with the contents of the page (the <UL><LI> structure is used to populate the menu, which is then displayed instead of the links).

The "display:none;" attribute on the 1st <UL> tag is used additionally, so that the <UL><LI> structure is hidden for a period of some milliseconds before AllWebMenus detects it and hides it with its own method.

While it is good to keep this attribute so that you do not see the structure "appear/disappear" for some milliseconds, you may decide not to keep it if you wish your links to be visible on browsers that do not support javascript (or have it disabled). Keep in mind however that this way the structure will always be visible until the menu starts loading (in the majority of users that **do** have javascript on their browsers).

4.6.3.1.3 Compile Properties: Menu Genre (at "Menu Editor" interface)

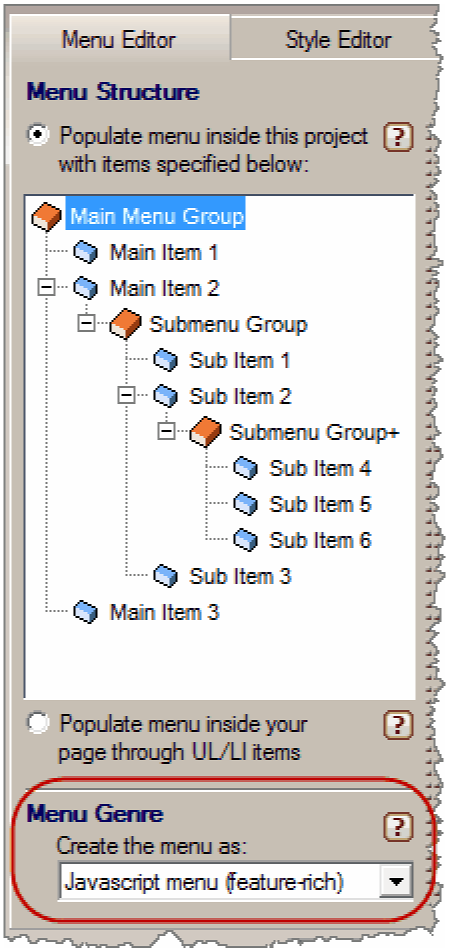
Here you specify how your menu functions:

- as a **"Javascript menu"** (feature-rich, all available features/options in AllWebMenus supported)

OR

- as a **"CSS menu"** (light, Section 508/WCAG compliant, works even with javascript disabled, not all features/options supported)

This option is located at the "Menu Editor" tab of the AllWebMenus main interface:



Menu Structure

● Populate menu inside this project with items specified below:

○ Populate menu inside your page through UL/LI items

Menu Genre

Create the menu as: Javascript menu (feature-rich)

The **"Javascript menu" Genre** implementation is the default one. It is the one that uses all features and options of the AllWebMenus interface. While **feature-rich**, it has some technical limitations:

- it **cannot be Section 508/WCAG compliant**, i.e. screen readers cannot read the text in menu items, convert it to audio and play it to the visually impaired,

- it uses a little **"heavier" code** than the "CSS menu" one, due to the more features it supports (like submenu opening/closing effects etc.),

- it requires that **javascript is enabled** on the browser (typical nowadays).

Contrary to the "Javascript menu", the **"CSS menu" Genre implementation** is designed so that:

- it is **Section 508/WCAG compliant** (required by certain governmental agencies),

- it uses **"lighter" code**, due to the less features it supports (lack of submenu opening/closing effects etc.),

- it is still functional when the **javascript is disabled** on the browser (degrades gracefully and still functions as a menu).

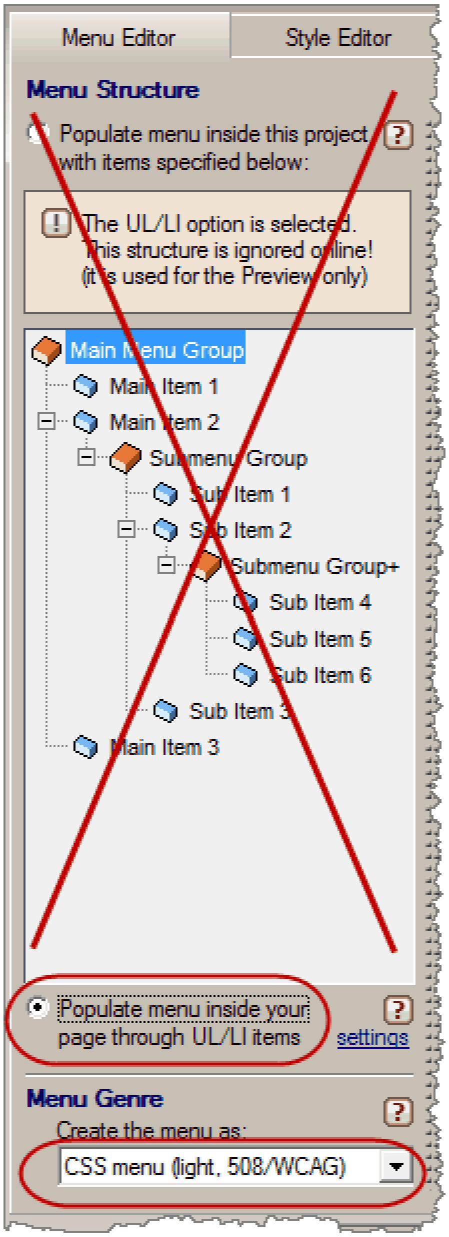
If you choose to use the "CSS menu" Genre implementation for any of the above **advantages**, just be aware that some features and options of the default "Javascript menu" cannot be supported (due to its lighter code and limited javascript).

Below is a list of all properties and their applicability (or not) in "CSS menu" implementations:

General features and "CSS Menu":

Menu Structure

The **"CSS menu" Genre implementation** requires that you use the "UL/LI" population method only, so that search engines and screen readers can parse the menu items (which are the "LI" elements):



○ Populate menu inside this project with items specified below:

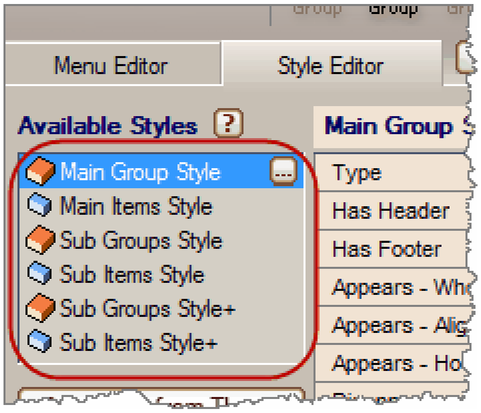
The UL/LI option is selected. This structure is ignored online! (it is used for the Preview only)

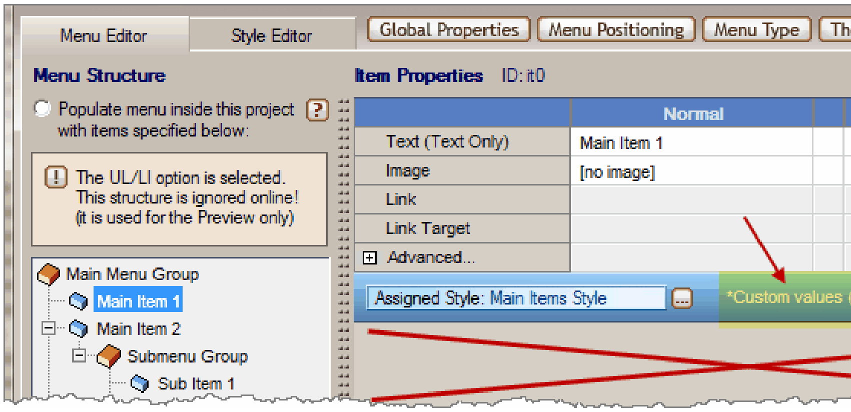
● Populate menu inside your page through UL/LI items

Create the menu as: CSS menu (light, 508/WCAG)

[Available Styles]

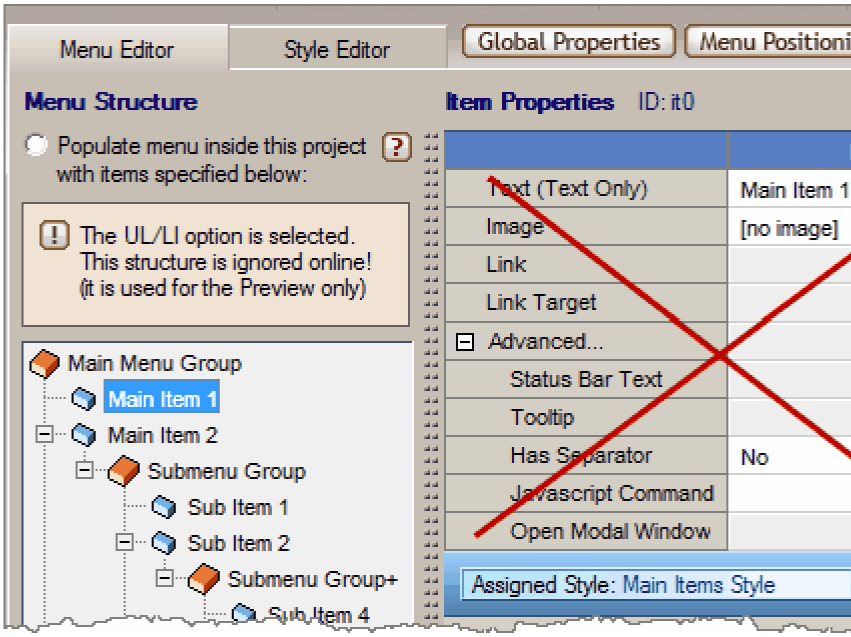
A "CSS menu" uses the 6 styles of the Style Editor for its appearance. No style customizations can be specified per item (something that you can do with a "Javascript menu"), since the "CSS menu" uses the "UL/LI" population method only. In that population method the style of each item ("LI" element) is based on its level on the menu: "Main item", OR "Submenu Item (2nd level)", OR "Submenu Item+ (3rd level and beyond)".





Content Properties

Similarly to the "custom Style values", none of the content properties is used by the online menu since this is populated by the "LI" elements on page instead (the preview does use them though, just for previewing purposes).



Mouse Click state	Not applicable. The values of the "Mouse Over" state is used when the user clicks on an item.

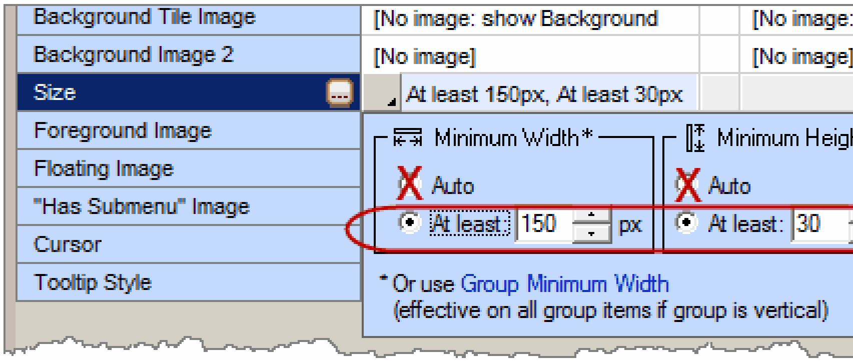
Group Style Properties and "CSS Menu":

Assigned Style	Applicable, you can assign one of the 3 generic group styles. However, no customizations per individual group are applicable.
Type	Only the "Horizontal/Vertical" option is applicable. Everything else does not apply to the online menu. This means that none of the options contained in the "Horizontal" or "Vertical" tabs are applicable, as well as the "Sliding" option. A CSS menu has "Uniform Items" by default, so this option is not applicable too.
Has Header	Applicable
Has Footer	Applicable
Appears - Where (submenus only)	Semi-applicable. The "edge" option is not applicable (submenus always appear at the right edge of parent group, or bottom edge if parent group is horizontal). The "distance" option however is applicable.
Appears - Aligned with (submenus only)	Semi-applicable. The "section" and "edge" options are not applicable (submenus always align with the right edge of parent item, or bottom edge if parent item is horizontal). The "alignment offset" option however is applicable.
Appears - How	Not applicable. Submenus always appear immediately, "on Mouse Over", without effects.
Disappears - How	Not applicable. Submenus are always non-sticky and disappear without effects.
Opacity	Applicable. Some browsers (IE) cascade the CSS opacity value to all the following submenu levels, so it is recommended to use opacity only at the Main level.
Background Color	Applicable.
Background Tile Image	Applicable.
Padding	Applicable.
Border	Applicable.

Rounded Corners	Applicable.
Distance Between Items	Applicable.
Width	Not applicable. The width of the group is determined by the "Item Size" property of its items.

Item Style Properties and "CSS Menu":

Assigned Style	Applicable, you can assign one of the 3 generic item styles. However, no customizations per individual item are applicable.
Content Alignment	Applicable.
Text Font	Applicable.
Text Size	Applicable.
Text Decoration	Applicable.
Text Color	Applicable.
Padding	Applicable.
Text Margin	Not applicable.
Border	Applicable.
Rounded Corners	Applicable.
Background Color	Applicable.
Background Image 1	Applicable.
Background Tile Image	Applicable.
Background Image 2	Applicable.
Size	Applicable (when non "Auto"). The "Auto" value is not supported by CSS (needs javascript), so both "width" and "height" needs to be explicitly specified in pixels. AllWebMenus will not let you compile if you forget these values to "Auto" in any of the 3 Item Styles, since a "CSS menu" cannot use javascript to calculate the size of all items on-the-fly.



Background Tile Image	[No image: show Background	[No image:
Background Image 2	[No image]	[No image]
Size	At least 150px, At least 30px	
Foreground Image		
Floating Image		
"Has Submenu" Image		
Cursor		
Tooltip Style		

Minimum Width*
Auto
● At least: 150 px

Minimum Heig
Auto
● At least: 30

* Or use Group Minimum Width
(effective on all group items if group is vertical)

Foreground Image	Not applicable.
Floating Image	Not applicable.
"Has Submenu" Image	Applicable.
Cursor	Not applicable.

Project Properties and "CSS Menu":

Folders & Files	Applicable. The "dot.gif" option is not applicable (not used in CSS menus).
Menu Structure Source	Applicable, only the "UL/LI" option.
Cross-Frame Support	Not applicable.
Optimization	Not applicable. All "Mouse Click" values are always ignored.
Menu-Linking Code in Tags	Not applicable.
FrontPage Support	Not applicable.
Advanced Scripting	Not applicable.
Custom Variables	Not applicable.
RTL Languages Support	Not applicable.
Menu Printing	Not applicable.

4.6.3.1.4 Compile Properties: SEO

Here you can generate code to use for search engine optimization purposes. You can apply any of the following two methods, or even both (combined):

Sitemap-based SEO

On-page SEO

Sitemap-based SEO:

Search engines will crawl your menu links through a **single sitemap file** that they find on your website.

Use the "Generic Sitemap" button to create a sitemap based on your menu structure.

A simple .html file will be exported that contains all your menu items along with their links (as <A> tags).

Then, just add a "sitemap" link on your index page (or any other page crawled by search engines) that links to that file and your most important pages of your website (those in your menu) will be indexed by the search engine crawlers, without any hassles!

Additionally, you may also use the "Google Sitemap" or "ROR Sitemap" to provide more info for certain search engines.

With this method (compared to the "on-page SEO" method) you do not need to add a lot of HTML code to your page (other than just adding a simple link to the sitemap file).

On-page SEO:

Search engines will crawl your menu links through **<UL><LI> HTML code** that you add to your pages.

Use the "View sample <UL><LI> code" button to see how your HTML code should be structured.

As you can see there, your menu links are presented as standard <A> tags.

Therefore, search engines will index them automatically whenever they crawl a page that contains the related <UL><LI> HTML code.

Example:

```
<ul id="awmULLI-menu" style="display:none;">
  <li><a href="index.html">Home</a></li>
  <li><a href="item1.html">Item 1</a></li>
  <li><a href="item2.html">Item 2</a>
    <ul>
      <li>Item 2.1
        <ul>
          <li><a href="item_211.html">Item 2.1.1</a></li>
          <li><a href="item_212.html">Item 2.1.2</a></li>
          <li><a href="item_213.html">Item 2.1.3</a></li>
          <li><a href="item_214.html">Item 2.1.4</a></li>
        </ul>
      </li>
      <li><a href="item_22.html">Item 2.2</a>
      </li>
    </ul>
  </li>
  <li><a href="item3.html">Item 3</a>
```

```
    </li>
  </ul>
```

Through the above <UL><LI> HTML, search engines will index all pages referenced by the <a> tags.

Notes:
- Some people believe that this method provides a little better SEO (compared to the "sitemap-based SEO"), as the links are followed directly from a major page rather than indirectly from a sitemap page. However, some other people believe that this method may not be as effective, since this list of links should be inevitably hidden from the user (through the "style" element on the first <UL>).
We provide you with both methods to choose the one (or both) that you believe will be most effective for your website.

- Using the "on-page SEO" method does not require you to also use the "<UL><LI> option" in the Menu Structure Source property. These are independent from each other. Actually, if you do so, your menu will be populated "on-the-fly" (based on the <UL><LI> code you use for SEO purposes), while you may not want this to happen. So, you can use the "Populate menu inside this project" option there and still use the "on page SEO" method.

4.6.3.1.5 Compile Properties: Cross-Frame Support

Menus and submenus are DHTML elements. Unfortunately, the DHTML specifications in some older browsers do **not** allow an object to be displayed outside its defined frame. Respectively, a menu is not able to open its submenus outside its assigned frame.

To account for this browser restriction we implemented a solution where the Main Menu appears on one frame and all its Submenus appear on another frame.

You can view an **on-line demo** on how to implement cross-frame menus at
http://www.likno.com/demos.html
You can also view an actual cross-frame implementation at
http://www.likno.com/examples/demo_frames.html

To use our solution on a framed web site you need to do the following steps:

STEP 1. Go to Tools-> Project Properties > Compile Properties and **check** the "Enable Cross-Frame Support" box. You will have to give two pieces of information:

"Frame Name where the Submenus will appear":
This is the **name** of the frame where the submenus will open (**not** the name of the html file that opens inside that frame!). A Wizard has been created to help you retrieve the proper name for your selected frame.

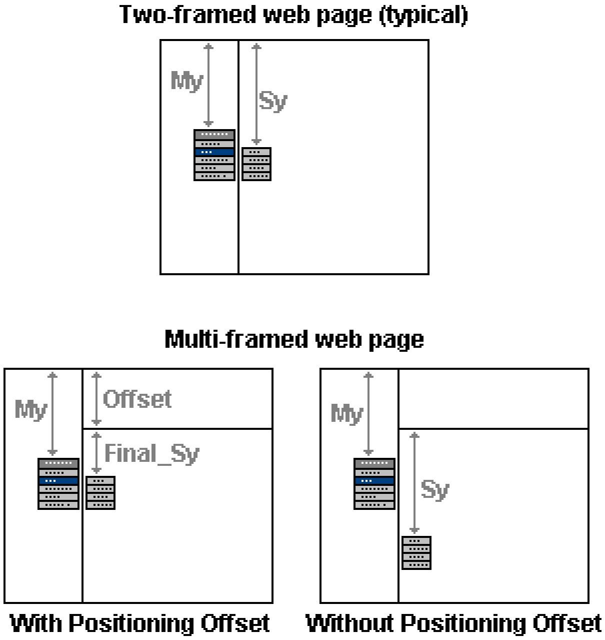
Example:
Let's assume the following Frameset Definition inside the source code of your home page:

```
<frameset framespacing="0" border="0" cols="171,*" frameborder="0">
  <frame name="left" target="main" src="left_page.htm" scrolling="auto">
  <frame name="main" src="main_page.htm" target="_self">
  <noframes>
  <body>
  <p>You need a browser that supports frames to see this site </p>
  </body>
  </noframes>
</frameset>
```

Let's also assume that you want the Main Menu to appear on the **left** frame and its submenus on the **main** frame.
The proper value for the field "Frame Name where the Submenus will open" should therefore be:
main (**not** *main_page.htm* !)

"Submenus Positioning Offset":
This field should only be used if your frameset has **more than two frames** and one of these frames affects the position of submenus relative to their Main Menu.

The best way to understand the functionality of this property is to read the following example:



Definitions:
My = vertical distance from top edge of Main Menu (in pixels)
Sy = vertical distance from top edge of the "Item 2" submenu (in pixels)
Final_Sy = vertical distance from **frame's** top edge of the submenu (in pixels)
Offset = the value you enter in the "Submenus Positioning Offset" field (positive or negative number of pixels)

In a two-framed web page, the submenu of "Item 2" opens correctly at Sy pixels from the top edge. The script automatically calculates the Sy distance based on the My value.

However, in a three-framed web page of the following setting, the submenu will open far away from its parent item due to the calculated Sy distance. The only way to avoid this is to manually enter the expected height of the top-right frame. In this particular setting, the Offset needs to be a **negative** number so that the **final** Sy distance is smaller.

The exact formula is: Final_Sy = Sy + Offset

STEP 2. Provided that the "Enable Cross-Frame Support" box is **checked**, the Compile Menu command creates **two** menu files instead of one:
the **menuname.js** file which contains the Main Menu only, and
the **menuname_sub.js** file which contains all the Submenus.

You need to link the *menuname.js* to the page where you want the Main Menu to appear, and the *menuname_sub.js* to the pages where you want the Submenus to appear.

> Example:
> Assuming that you use the following frameset:
> <frame name="**left**" target="main" src="**left_page.htm**" scrolling="auto">
> <frame name="**main**" src="**main_page.htm**" target="_self">
>
> **menuname.js** should be linked to the **left_page.htm**,
> **menuname_sub.js** should be linked to the **main_page.htm** and **all other pages** opening at the main frame.

Tip1: If you have only created frames so that your navigation is always visible, there is a better way to achieve this. **You can avoid having frames by using a "Floating" menu at the left or top of your page**.
To view an on-line example on how to avoid frames using the "Floating" feature, you may click on the link below:
http://www.likno.com/examples/example_frames_floating.html

Tip2: For "horizontal" menus you may use "horizontal submenus" and still use frames, allowing a little space for the horizontal submenus to open.

Important Notes!
1. Using the cross-frame feature for menus placed on right frames:
Submenus always open at the left side of the target frame. While this is normal behavior for menus placed on left frames, it is not as good for menus placed on right frames (eg: Arabic or Hebrew framed web sites).
If you have a right-to-left, framed web site, and the Main Menu is placed on the right frame, then you need to enter the following code into the "Advanced Scripting" property:

```
var awmRightToLeftFrame=2;
```

Only if you use this variable will the submenus appear at the right side of the left frame (target frame for the submenus). Otherwise, the submenus will appear at the left side of the left frame (default behavior).

2. Do not use html files with a system default name like **top.html**, as it appears that it conflicts with conventional default target names top, new, parent, etc...

4.6.3.1.6 Compile Properties: Optimization

This topic is described in detail in the "Optimize menus to improve building speed" section.

4.6.3.1.7 Compile Properties: Menu-Linking Code in Tags

In this Compile Property you inform AllWebMenus if the menu-linking code is a **direct child** of the <BODY> tag (default behavior) OR it "sits" inside additional tags (like <td>, <div>, <span>, etc.).

Why should you do this?

By default, the AllWebMenus "automatic linking" process inserts the menu-linking code at the beginning of the page's HTML main code, right **after** the <BODY> tag (and outside any other tags).

The same you should also do if you choose to add the code manually. This allows the menu to be generated in the background while the browser still loads the rest of the page, hence it appears fast.

However, depending on the CMS or website builder you use, you may not be able to insert the menu-linking code there.

In such a case:

- If the menu-linking code is placed differently but is still a **direct child** of the <BODY> tag, you do **not need to change** the default *"No, outside tags"* option.

- If the menu-linking code is placed differently but is **not a direct child** of the <BODY> tag (i.e. it is also **within other tags**), you need to set the option to *"Yes, within tags"*.
Otherwise, your menu **may appear broken** in certain browsers!

By using the "Yes, within tags" value, the menu engine starts building the menu only **after** the entire **page is fully loaded**. With this technique, the existence of any nested tags **do not break** the building and rendering of the menu.

The only minimal "effect" of this option is that your menu appears **after the page is fully loaded**, not as quickly as with the "No, outside tags" option. However, this is very minimal and unnoticeable in most cases, as the menu files are cached after visiting the first page that contains them.

Note on positioning:
- The menu-linking code does **not** determine where the menu appears on page (this is specified through the <Positioning> property). Therefore, moving it from right AFTER the <BODY> tag will have no effect on menu positioning.

Note on <UL><LI> case:
You may see this property **disabled**. This happens when the <Menu Structure Source> property is set to the "<UL><LI> items" value. Why? This value requires that the page must be **fully loaded** before the menu can be rendered on page. Therefore the "Menu-Linking Code in Tags" defaults to "Yes" (which has the "fully load" requirement), regardless of the selection made. It is enabled again only if you change the "Menu Structure Source" to the "Use project's structure" value.

See also
Menu-linking code

4.6.3.1.8 Compile Properties: FrontPage Support

If you are a FrontPage user you can link your menu into a Shared Border so that you won't have to link it into each page separately.

Note: You should link the menu **only to the Bottom Shared Border** to make sure that you won't get into trouble with certain browsers. The reason for this restriction is explained in the last section of this page.

How to use the Bottom Shared Border to display a menu in all pages:

1. Inside FrontPage, **check** the **Bottom** Shared Border box in all pages where you want the menu to appear.
(see notes below to understand why **you should not choose the top, left or right shared border** to insert the menu-linking code)

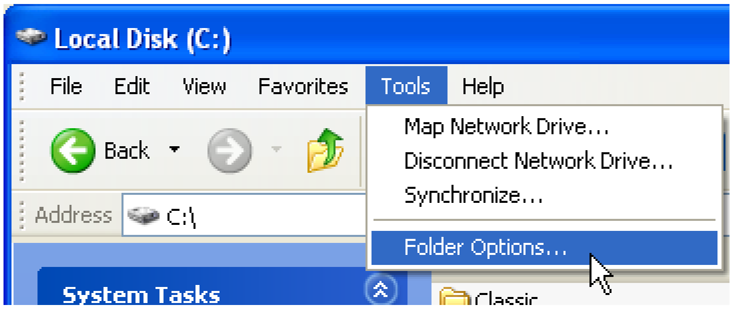
2. Inside AllWebMenus, go to Tools-> Project Properties > Compile Properties-> FrontPage Support and check the "Support Shared Borders in FrontPage" box.

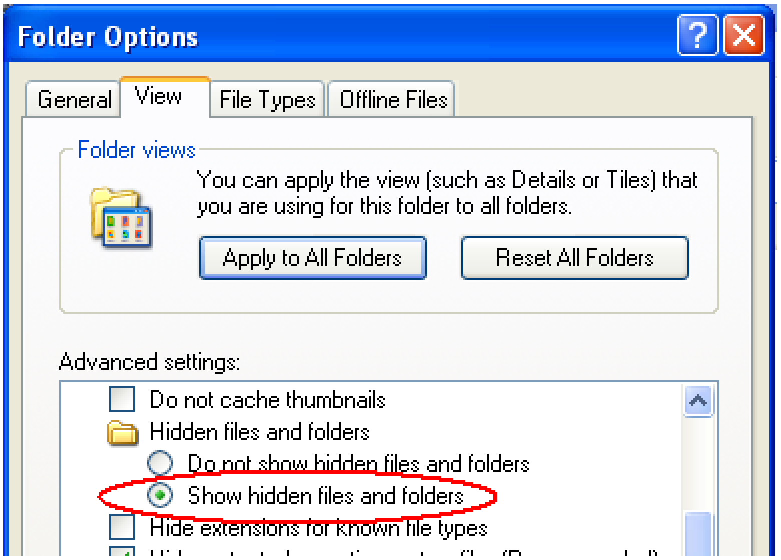
3. Close your Web in FrontPage so that all compile/link changes take effect.

4. Compile the menu at your WEB's **root** (**not** inside your _borders directory).

5. Use the "Link Compiled Menu to Web Page(s)" command to link the menu to your WEB's ***_borders/bottom.htm*** file.

Note! The _*borders* directory is set to be **hidden** by Frontpage. In order to work with the files in this directory you will need to change a setting in your "Folder Options" of your Windows Explorer, as illustrated below:





You can apply the view (such as Details or Tiles) that you are using for this folder to all folders.

□ Do not cache thumbnails
Hidden files and folders
○ Do not show hidden files and folders
● Show hidden files and folders
□ Hide extensions for known file types

Following the steps above, the menu appears in all pages containing the Bottom shared border.

The exact positioning of the menu is determined by other properties (see paragraph below).

(note: the presence of the menu-linking code inside the Bottom shared border does not necessarily mean that the menu appears in all pages containing the Bottom shared border. You may want some of these pages not to show the menu. To achieve this behaviour, you should compile the menu using the "Element-relative positioning" method on the <Positioning> property and make sure that you do not include the menu-positioning code on these pages. With this way the menu will not appear on the page.)

Why linking the menu to the Bottom Shared Border does not affect the desired position of the menu on the page:

Since the menu **floats** on the page, its position is not determined by where you place the AllWebMenus menu-linking code (i.e. the code inserted when using the "Link Compiled menu" command). Instead, its position is determined by the <Vertical Distance from Edge> and <Horizontal Distance from Edge> properties, **or** by the **menu-positioning code** that you may have inserted at the desired location on page. So, position-wise it does not matter **where** you place the menu-linking code (where you link the menu). However, for cross-browser reasons it **does** matter that you **not move** this code from the end of the page! And the only way to achieve this with Shared Borders is to link the menu into the **Bottom** Shared Border and check the related box in Compile Properties.

For more info on how to achieve exact menu positioning by using the related menu-positioning code you may read the "How Do I Position the menu relative to a page element" section.

See also
How Do I Position the menu relative to a page element
Compile Menu
How Do I Use my designed menu on my web pages

4.6.3.1.9 Compile Properties: Advanced Scripting

In this Compile Property you can specify additional javascript code to be inserted into the menu.js file upon compilation of the menu.

This additional code will be executed every time the menu appears on page (i.e. every time the menu srcipt is running).

An experienced developer can integrate several javascript effects inside the menu.js file. You can imagine that this extends the AllWebMenus functionality dramatically, since the menu.js runs in all the pages of your web site.

Your scripts may produce javascript errors in the Preview pane if they rely on information located at your actual web pages. In this case it is recommended that you check the "Ignore script in Preview pane" so that they do not run at preview mode. The scripts ***do*** run on your actual web pages.

Note 1: Do not use the <script> & </script> tags. The menu.js is already a javascript file.

Note 2: This feature is for advanced users only! You may produce Javascript Errors if you do not include valid code here.

4.6.3.1.10 Compile Properties: Custom Variables

In this Compile Property you can specify default strings for any custom variable (eg: **`<awm:myvar>`**) you have added to applicable properties. See examples below or an additional example online.

A custom variable is a **javascript variable** whose value you can update in your web page (through javascript) in order to alter the menu properties where they are used, based on your conditions or preferences.

- The syntax of a custom variable is (whichever you prefer):

  <awm:myvar>

  OR

  <awm:myvar/>

- The applicable properties where a custom variable can be used are:

  ITEM PROPERTIES:

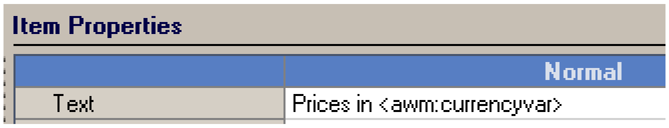
  <Text>
  <Link>
  <Status Bar Text>
  <Tooltip>

  GROUP PROPERTIES:

  <Header Text>
  <Header Tooltip>
  <Footer Text>
  <Footer Tooltip>

- **Example** on how a custom variable is used:

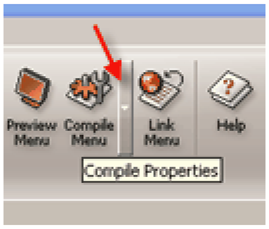
  Suppose you add a custom variable, called "currencyvar", in the <Text> property of an item, using the following syntax:



Item Properties

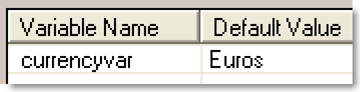
	Normal
Text	Prices in <awm:currencyvar>

You then click on the "Tools -> Project Properties" menu OR the "Compile Properties" toolbar button:



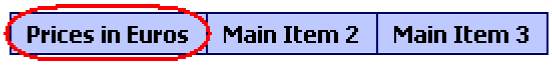
You click on the "Custom Variables" option and see the "currencyvar" variable already **detected** and added to the list.

Suppose you set the **default** value of the variable to "Euros":



Variable Name	Default Value
currencyvar	Euros

This will result to the following text on the menu item by default:



Now, suppose that you are using PHP (same applies to any other language).

You can specify a different value for the "currencyvar" javascript variable through the following sample code:

```
<body>
<?
            $country = $_GET['country'];
            switch ($country) {
                        case "EU":
                        case "Germany":
                        case "Greece":
                        case "France":
                        case "Spain":
                        default:
                                    $cur = "Euros";
                                    break;
                        case "UK":
                                    $cur = "British Pounds";
                                    break;
                        case "US":
                                    $cur = "US Dollars";
                                    break;
            }
               //This custom variable used in the AllWebMenus project
takes a value *before* the menu-linking code builds the menu.
               //Otherwise, the "default" value of the variable will
be shown.
               //The below line sets the value to the custom
javascript variable:
```

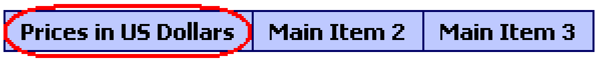
```
            echo "<script type='text/javascript'>var currencyvar =
'$cur';</script>";

?>

<!-- ******** BEGIN ALLWEBMENUS CODE FOR menu ******** -->
<script type="text/javascript">var MenuLinkedBy="AllWebMenus
[4]",awmMenuName="menu",awmBN="628";awmAltUrl="";</script>
<script charset="UTF-8" src="menu.js"
type="text/javascript"></script><script
type="text/javascript">awmBuildMenu();</script>
<!-- ******** END ALLWEBMENUS CODE FOR menu ******** -->

<!-- HTML of the page continues below... -->
```

Assuming that the $country is "US", the following text will appear on the menu item:



In general, you can specify the value of any "custom variable" used in the AllWebMenus project through **javascript code in your page**:

```
<script type='text/javascript'>
var variable1 = value1;
var variable2 = value2;
</script>
```

You can see an additional example online.

Note 1: The list of custom variables is **automatically populated (detected)** through "smart crawling" of the applicable menu properties. You cannot add or delete variables through the "Custom Variables" form, as this is done automatically by the crawling process.

Note 2: A custom variable in the <Text> property will NOT change the initially rendered size of the menu item, so your menu will seem "broken" if your custom variable results in a text size larger than the item can fit.

Note 3: Updating the custom variable in the <Text> property of an item while this item is already visible on page will have **no effect** (for example when it is a "Main Menu" item OR a submenu item of an opened sliding submenu). Still, there is a way to **"refresh" the menu** after a custom variable related to the <Text> property has been updated, by calling a special javascript function: `awm_refresh_custom_variables();`

For example:

```
<script type='text/javascript'>
  var variable1 = newvalue1;
  var variable2 = newvalue2;
  awm_refresh_custom_variables();
</script>
```

Note that if in your page you do not specify values for the custom variables (with javascript) **prior to the menu-linking code**, the **default** values of these variables will appear instead (i.e.

the ones specified through the currently explained feature of AllWebMenus).

The other properties (<Link>, <Status Bar Text>, <Tooltip>) do not need the function `awm_refresh_custom_variables();` or any other special treatment, as their custom variables are "parsed" on-the-fly (i.e. "on action").

Note 4: This feature is for advanced users only! You may produce Javascript Errors if you do not use valid syntax in custom variables. No support will be provided on mishandling of javascript variables.

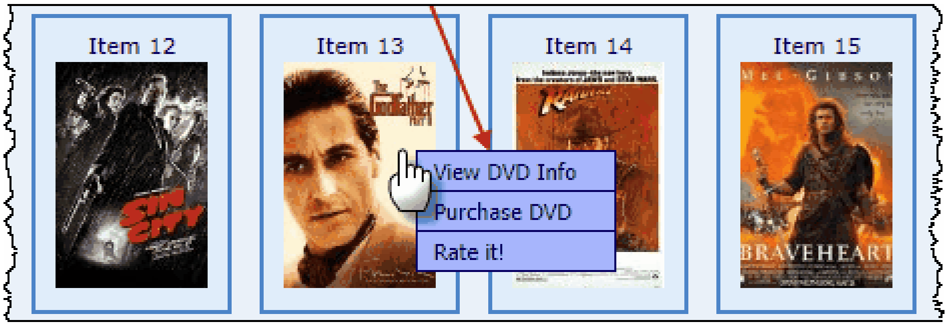
Examples

Example 1: See "Menu Variables Example" online

Example 2: Read below how you can show the **same single menu** on different page elements, **using different <Link> values** on its items per element.

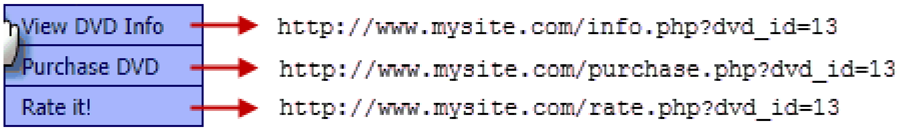
Let's assume that you have several images on your page that display different DVD movie posters.

Each of these images shows the same menu when clicking on it, offering 3 actions (items) for the particular DVD that opens it ("view it", "purchase it", "rate it", etc.):



Now, each item of that menu links to a page related to its action (e.g. the "Purchase DVD" item links to the "purchase.php" page, etc.). In order for the linked page to know which DVD opens it, there is also a parameter at the end of the URL (e.g. "?dvd_id=13").

For example, these are the menu links when "Item 13" (DVD 13) triggers the menu:

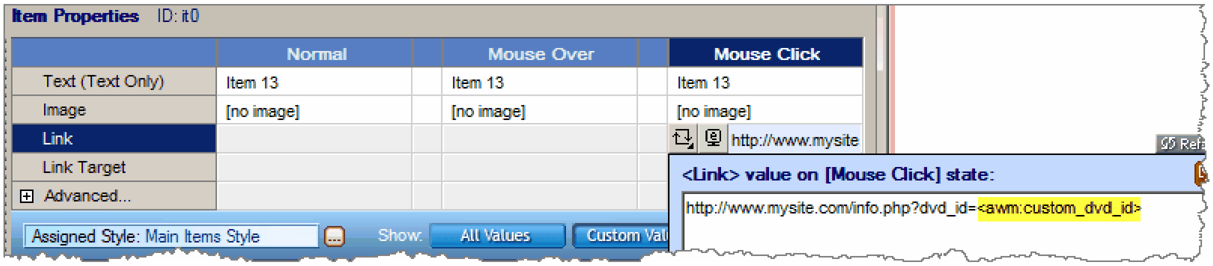


```
http://www.mysite.com/info.php?dvd_id=13
http://www.mysite.com/purchase.php?dvd_id=13
http://www.mysite.com/rate.php?dvd_id=13
```

But how can you use a **different link value** on the **same menu**, depending on which image (page element) triggers its appearance?

This is how:

1. In AllWebMenus, use a **custom variable** at the final part of each item's <Link> property (instead of a "hard coded" value):



Item Properties ID: it0

	Normal	Mouse Over	Mouse Click
Text (Text Only)	Item 13	Item 13	Item 13
Image	[no image]	[no image]	[no image]
Link			http://www.mysite
Link Target			
⊞ Advanced...			

Assigned Style: Main Items Style

<Link> value on [Mouse Click] state:

http://www.mysite.com/info.php?dvd_id=<awm:custom_dvd_id>

2. In your page, first **specify the value** of this custom variable (with javascript) and then **manually trigger** the appearance of the menu (also with javascript):

```
<img src="movie-poster-13.jpg" onclick="custom_dvd_id=13;
awmShowGroup('menu-gr0',0,3,0,0,this);">
```

For each different page element (DVD image) use a different value prior to calling the same menu. For example:

```
<img src="movie-poster-14.jpg" onclick="custom_dvd_id=14;
awmShowGroup('menu-gr0',0,3,0,0,this);">
<img src="movie-poster-15.jpg" onclick="custom_dvd_id=15;
awmShowGroup('menu-gr0',0,3,0,0,this);">
<img src="movie-poster-16.jpg" onclick="custom_dvd_id=16;
awmShowGroup('menu-gr0',0,3,0,0,this);">
```

In that way, the menu links will use a different value in their URL parameters depending on which page element (DVD image) opens it.

For more information on how to **manually trigger a menu** (on demand, as a "popup menu") please read the "Manually trigger Main Menu or any Group on user action" positioning method.

4.6.3.1.11 Compile Properties: RTL Languages Support

Here you can specify your menu to be compiled for Right-to-Left (eg: Arabic, Hebrew) web sites.

Just check this option and your menu will automatically function as a Right-to-Left menu:

- All submenus of an horizontal menu will align to the **right** edge of the parent item.

- The "Auto" value of the <Appears - Where> property makes the submenu appear to the **right** side of the parent group

- Cross-frame implementations make the submenus appear on the **right** side of the frame that contains the submenus, instead of left. (Note: if you want to have RTL support, but still want your submenus to appear on the left side of the frame that contains them, then you need to enter the following code into the "Advanced Scripting" property: var awmRightToLeftFrame=2;

4.6.3.1.12 Compile Properties: Menu Printing

When printing a page, its menu layout may be misplaced as browsers do not support accurate printing of DHTML elements (any type, not only menus).

If you want to overcome this, you can have the related option checked: "Do not print the menu layout when printing the page"

4.6.3.1.13 Compile Properties: Link & Run Other Likno Projects

Here you can link (and run) **other Likno Projects** (for example a "Likno Tooltips Project") on the same pages where this menu is also linked to.

This is like adding the "linking code" of such Likno Projects on the same pages (but without actually doing so).

When is this useful?

This is useful when you want a Likno Project to run where the menu also appears (most likely the entire website) and **you want to skip the "hassle" of separately linking it** to the same pages through its own "linking process".

Example:

> You create a stylish tooltip with "Likno Web Tooltips Builder" to show the content of the "ALT" attribute of each image for your entire website.
> You can either link its compiled project on all pages through the "Likno Web Tooltips Builder" OR just add it to this list and forget about linking it separately. No adding of the tooltips "linking code" to the pages is needed if you choose the 2nd option.
>
> Read how we use this feature on our own website (Use Case)

When is this NOT a good idea?

You will cause unnecessary overhead to all pages if this Likno Project is intended to run only on a particular page or just a few pages. In such a case just use the "linking process" of the Likno Product used to create it.

Note: This feature does not apply to "Joomla, Wordpress or Drupal" menu compilations (our Add-ins have their own way of linking such Likno Projects on websites created with them).

4.6.3.2 Preview Panel Settings

You can specify options for the Preview Panel, such as the Background Color, the text Encoding, the Doctype and if the menu is automatically refreshed when a change is made to a property.

Notes:
- The "Automatically Refresh Preview" is not recommended if you have very large menus as it may slow you down when editing the menu properties.

- Both the “Doctype” and “Encoding” values are used in previewing the menu **within AllWebMenus** and are not applied to your actual pages. Your pages will have their own HTML code for Doctype and Encoding and therefore the menu will be rendered according to those values; NOT according to the Menu Preview values. The Menu Preview options are there just to facilitate the preview. Read here more

4.6.3.3 Project Info Settings

Here you can specify settings for your menu project, such as:

Author Name
Author URL
Project Description
Design Notes

4.6.4 Command: Link Compiled Menu to Web Pages

Links a compiled menu (.js file) to a selection of web pages.

At this stage, AllWebMenus inserts the necessary menu link into the selected web pages. No DHTML or Javascript knowledge is required since the whole process is conducted by the application. The menu link is a few lines of code and can be easily identified inside the source code by a related comment.

The linking process requires only two pieces of information:
1. The **compiled menu file** (eg: mymenu.js)
2. The **web page(s)** where the menu link will be created.

Obviously, you need to **upload** the updated web pages on the server so that the menu can be viewed on-line. However, it is important that you also upload...

1. the compiled menu file (eg: ***mymenu.js***).
2. the awmdata folder.

...at the **same** directory structure where they reside locally.
See the Compile Menu command for more info on this issue.

To update the design of an already compiled menu you should first make the changes to its respective **.awm** file and then recompile it. Provided that you did not change the name or the directory of the compiled menu, you do not need to re-link the menu to the pages that already contain that link.

Note: If you are using the "Position relative to Element" Positioning option, please make sure that you have the "Element ID" added to the web page, otherwise the menu will not show up.

How does the menu-linking process work:
Let's say that you want to link a menu called "menu3.js" to a page. The process asks the following question and takes the respective action:

Does menu-linking code already exists for "menu3.js"?
Yes -> Replace the old menu-linking code with the new one
No -> Insert a **new** menu-linking code for "menu3.js" into the page

Notes:
Note about file extensions:
The link can be inserted in any kind of page, regardless of its extension. Just use the All Files (*.*) filter to select web pages that have an extension other than the typical .html or .htm one.

Note for Dreamweaver users:
If you like, you can also link the compiled menu to a template (.dwt) or a library (.lbi). Just select the related directory (Templates or Library) and use the All Files (*.*) filter so that you can view the template or library file.

The library (.lbi) is a special Dreamweaver file containing a collection of individual assets or copies of assets that you can place in your web pages. That means the menu linking code can be added in a library but you should make sure that the library should be placed as the first element after the BODY opening tag (<body>) and that the linking code is the first element, listed in the library file itself.

The menu does not have to be compiled inside the Templates directory. On the contrary, we do not advise you to do so as you will have to upload the Templates directory on the server too.

Note for FrontPage users:
Similarly to Dreamweaver, you can link your menu into an **include** so that it can be used by all other pages. Just make sure that this include contains only the menu and that it is inserted immediately before the </body> tag of the pages. To use the menu inside a Shared Border, please read the "How Do I Link menus to FrontPage Shared Borders" section.

Note for users of the "WordPress Menu Add-in", "Joomla Menu Add-in", etc.:
The menu-linking code is **different on these environments!**
Check the online section of each Add-in for instructions on its different menu-linking code.
You may also find links on pages describing the different menu-linking code of several Add-ins here.

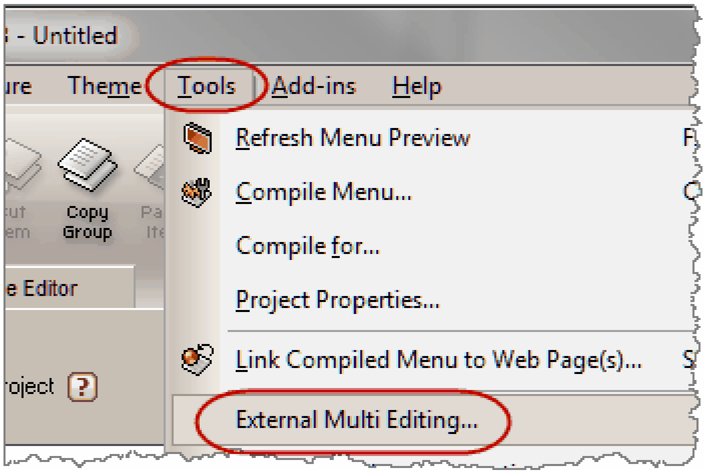
See also
Compile Menu
The alternative web page
How Do I Position the menu relative to a page element
How Do I Link menus to FrontPage Shared Borders
How Do I Use my designed menu on my web pages

4.6.5 Command: External Multi Editing

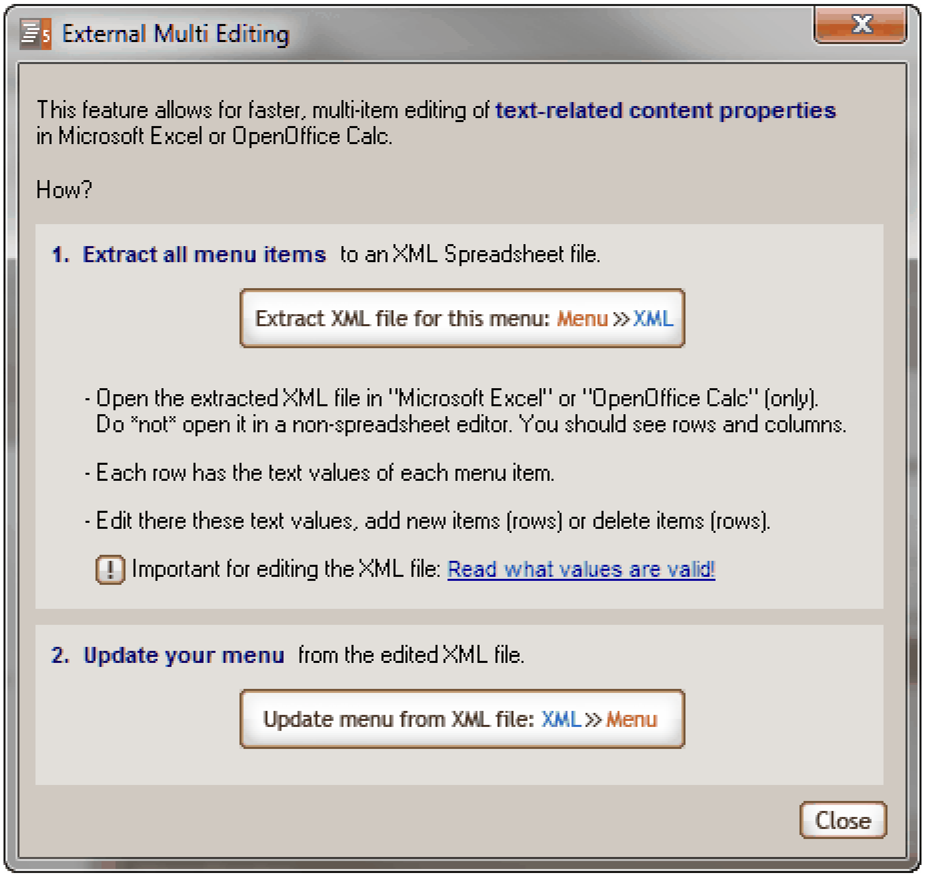
You can use this feature **for editing** certain text-related properties of an AllWebMenus project (.awm) in Excel or OpenOffice Calc.

How?

Choose the "External Multi Editing" command in the "Tools" menu:



Then, click the "*Extract XML file for this menu*" button.


External Multi Editing
Extract XML file for this menu: Menu >> XML
Update menu from XML file: XML >> Menu
Close

This feature allows for faster, multi-item editing of **text-related content properties** in Microsoft Excel or OpenOffice Calc.

How?

1. Extract all menu items to an XML Spreadsheet file.

- Open the extracted XML file in "Microsoft Excel" or "OpenOffice Calc" (only). Do *not* open it in a non-spreadsheet editor. You should see rows and columns.

- Each row has the text values of each menu item.

- Edit there these text values, add new items (rows) or delete items (rows).

! Important for editing the XML file: Read what values are valid!

2. Update your menu from the edited XML file.

This produces an XML file with specific properties of the menu, which you then edit in Excel.

After you finish editing the supported menu properties in the XML file, save it and click the "*Update menu from XML file*" button. This opens an "Open XML Spreadsheet File" dialog. Select your saved XML there and click "Open".

This is it!
Your menu now gets the updated values that you edited in Excel.

Note that this is not a *"do-it-all-in-Excel"* feature. You can edit some particular properties only.

For example, you cannot edit the "Item Image" or "Item Separator" properties, you cannot edit the style properties, etc.

This feature can be used for **easy search and replace** on text-related properties (such as changing the URL on all links), by editing the XML file in Excel and then updating your AllWebMenus project accordingly.

Still, you can also **add** or **delete rows**, thus performing some menu structure management outside of the AllWebMenus interface. This is an advanced capability (read below how this can be done) so **use it with care!**

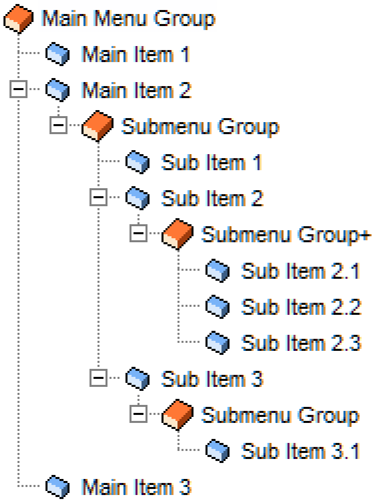
We will **NOT offer support for errors in your XML.** Saving your project before updating it with the XML file is strongly recommended, as the update removes your previous menu structure totally.

Below is an explanation of what each column in Excel represents and what its values are:

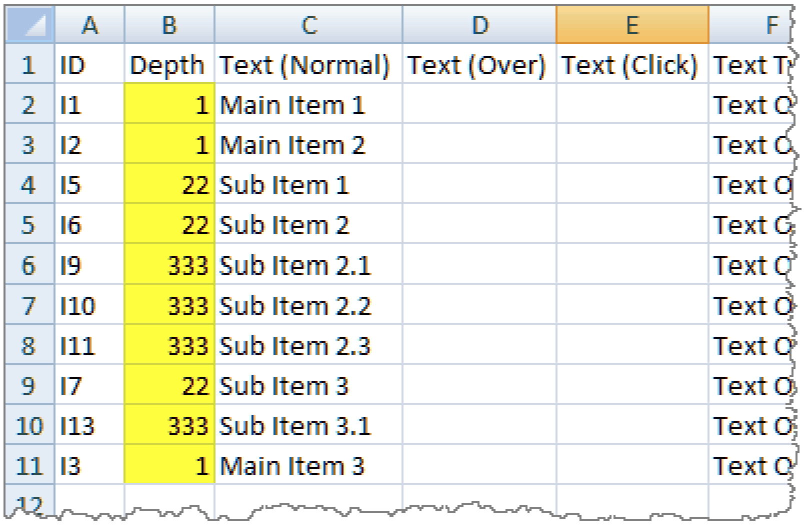
"Depth" column

The "Depth" shows if an item is 1st level, 2nd level, 3rd level, etc.

For example, for this menu...



...these are the "Depth" values in the XML file (as edited in Excel):



	A	B	C	D	E	F
1	ID	Depth	Text (Normal)	Text (Over)	Text (Click)	Text T
2	I1	1	Main Item 1			Text O
3	I2	1	Main Item 2			Text O
4	I5	22	Sub Item 1			Text O
5	I6	22	Sub Item 2			Text O
6	I9	333	Sub Item 2.1			Text O
7	I10	333	Sub Item 2.2			Text O
8	I11	333	Sub Item 2.3			Text O
9	I7	22	Sub Item 3			Text O
10	I13	333	Sub Item 3.1			Text O
11	I3	1	Main Item 3			Text O
12						

1 means that the item is a Main Menu item (1st level)
22 means it is a submenu item of a Main Menu item (2nd level)
333 means that it is a submenu item of another submenu item (3rd level)
and so on...

If you edit the XML file so that you **just change the text** on the **existing items**, there is no need to know anything else about this column. Just don't touch its values and you will be OK.

However, if you edit the XML file so that you **also add** or **remove items** (rows) from your menu structure, then please read below **very carefully**, as only specific values can be used in the cells of this column!

The number of characters should **always match the row's level**:

1	Correct
22	Correct (two "2" characters)
333	Correct (three "3" characters)
4444	Correct (four "4" characters)
4	Incorrect
444	Incorrect
55555	Correct (five "5" characters)
5	Incorrect

If you have more than 9 levels (very rare), you should use the alphabet characters:

AAAAAAAAAA (10 "A"s represent the 10th level)
BBBBBBBBBBB (11 "B"s represent the 11th level)
and so on...

The following rules apply:

The level of the first menu item (row) must always be "1".

A submenu item should always be a child of a "previous level" item. This means that a "1" row can only be followed by another "1" row (same level item) or a "22" row (submenu item), but not a "333" row or a "4444" row (this would make no sense).

The "level" of an item (row) compared to the level of its previous item can be:

- the **same**, if the two items are **siblings**.

- **bigger by one** (and only one!), if it is a **child** of the previous item.
  In this case the previous item has a submenu group and the current item is the first in this submenu group.

- **smaller**, if it is a member of a group which is "parent" (or any "grandparent") of the group where the previous item belongs.

For example, after a "22" row you can have any of the following:

- another "22" row (another 2nd level item in the same submenu)

OR

- a "333" row (a submenu item for this item)

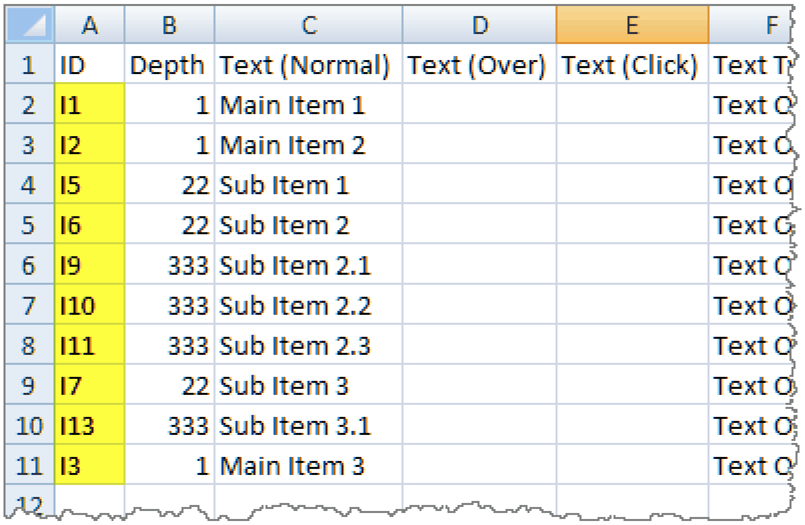
OR

- a "1" row (no more submenu items, next item is a 1st level item)

OR

nothing (the menu structure just ends there)

"ID" column



	A	B	C	D	E	F
1	ID	Depth	Text (Normal)	Text (Over)	Text (Click)	Text T
2	I1	1	Main Item 1			Text O
3	I2	1	Main Item 2			Text O
4	I5	22	Sub Item 1			Text O
5	I6	22	Sub Item 2			Text O
6	I9	333	Sub Item 2.1			Text O
7	I10	333	Sub Item 2.2			Text O
8	I11	333	Sub Item 2.3			Text O
9	I7	22	Sub Item 3			Text O
10	I13	333	Sub Item 3.1			Text O
11	I3	1	Main Item 3			Text O
12						

These IDs are used by AllWebMenus when performing the *"Update menu from the XML file"* process (opposite of the *"Extract XML file for this menu"* process).

Using these IDs, AllWebMenus can retain the style information of your existing menu items in the project when the XML info is applied.

It is important to follow these rules, otherwise the *"Update menu..."* process will break:

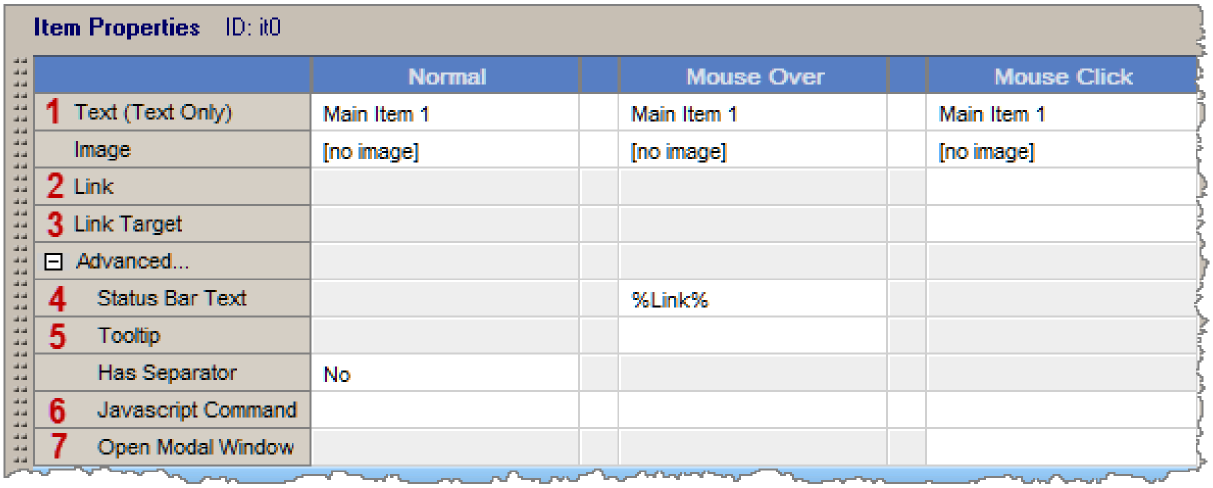
- **Do not edit these values!**
  Your items in the AllWebMenus project from which the XML is extracted need these values to "synchronize" some of their properties that cannot be edited through the XML file (such as style, Image, Has Separator, etc.).

- When you add **new** items (by adding new rows), make sure that their ID cell is **blank**!
  In other words, **new items should always have a blank ID.**

- If you duplicate (copy/paste) an existing item to create a new item, the new item will get the same ID value. In this case, you need to remove this value and make it **blank**!

"Text" columns (3 columns per state)

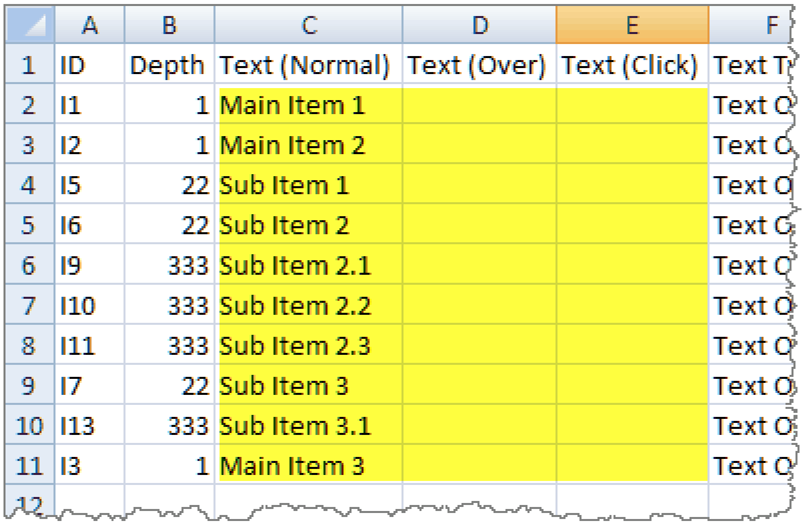
The cell values of these columns refer to the "Text" AllWebMenus values (**"1"** in screenshot below).



Item Properties ID: it0

	Normal		Mouse Over		Mouse Click
1 Text (Text Only)	Main Item 1		Main Item 1		Main Item 1
Image	[no image]		[no image]		[no image]
2 Link					
3 Link Target					
⊟ Advanced...					
4 Status Bar Text			%Link%		
5 Tooltip					
Has Separator	No				
6 Javascript Command					
7 Open Modal Window					

Just add/edit the item's text in these cells.



	A	B	C	D	E	F
1	ID	Depth	Text (Normal)	Text (Over)	Text (Click)	Text T
2	I1	1	Main Item 1			Text O
3	I2	1	Main Item 2			Text O
4	I5	22	Sub Item 1			Text O
5	I6	22	Sub Item 2			Text O
6	I9	333	Sub Item 2.1			Text O
7	I10	333	Sub Item 2.2			Text O
8	I11	333	Sub Item 2.3			Text O
9	I7	22	Sub Item 3			Text O
10	I13	333	Sub Item 3.1			Text O
11	I3	1	Main Item 3			Text O
12						

For most projects, the **"Text (Over)"** and **"Text (Click)"** cells will be left blank. Only if you want them to be **different than the "Text (Normal)"** cell, you should use values there.

Actually, the following rules apply when the "Update menu..." process takes place.

- If "Text (Over)" cell is blank: the "Text (Normal)" value is applied to the AllWebMenus "Mouse Over" state for this item.

- If "Text (Click)" cell is blank: the "Text (Over)" value is applied to the AllWebMenus "Mouse Click" state for this item.

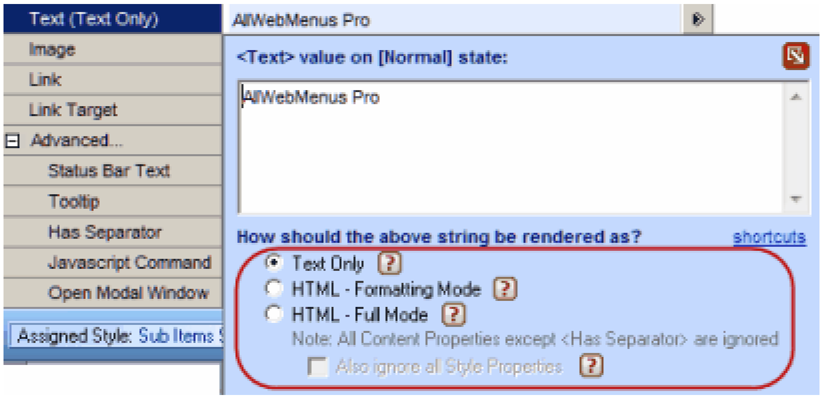
- If both the "Text (Over)" and "Text (Click)" cells are blank: the "Text (Normal)" value is applied to all "Text" AllWebMenus states for this item.

- If you want a "Text (Over)" or "Text (Click)" cell to be applied literally as "blank" (empty) in AllWebMenus, you should use the following special token in the cell: **[no-text]**

More info on the "Text" property.

"Text Type" column

The cell value of this column refers to the *"How should the above string be rendered as?"* section of the "Text" AllWebMenus property in AllWebMenus.



Text (Text Only)
Image
Link
Link Target
⊟ Advanced...
Status Bar Text
Tooltip
Has Separator
Javascript Command
Open Modal Window
Assigned Style: Sub Items S

AllWebMenus Pro

<Text> value on [Normal] state:

AllWebMenus Pro

How should the above string be rendered as? shortcuts

● Text Only ?
○ HTML - Formatting Mode ?
○ HTML - Full Mode ?
Note: All Content Properties except <Has Separator> are ignored
□ Also ignore all Style Properties ?

Valid values:

For the default "Text Only" value, **just leave this cell blank**. This is the most common case.

If you want a different option for an item, these are the **exact** values you are allowed to use on its "Text Type" cell:

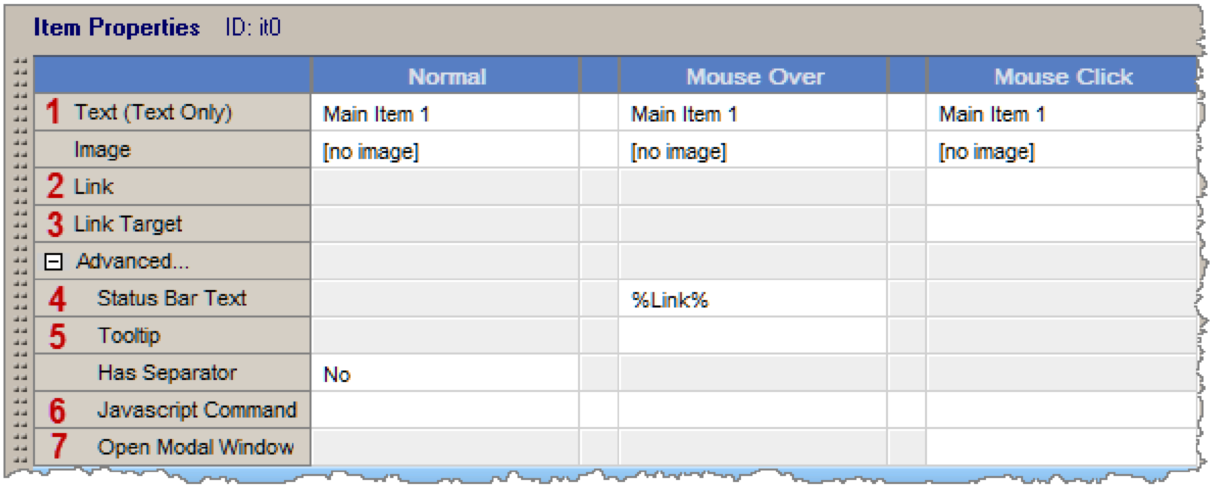
HTML - Formatting Mode
HTML - Full Mode
HTML - Full Mode - Ignore Style

Note: If a "Text Type" cell uses a misspelled value from the above, the *"Update menu..."* process will break!

More info on the "Text" property and its types.

"Link" column

The cell value of this column refers to the "Link" AllWebMenus value (**"2"** in screenshot below).



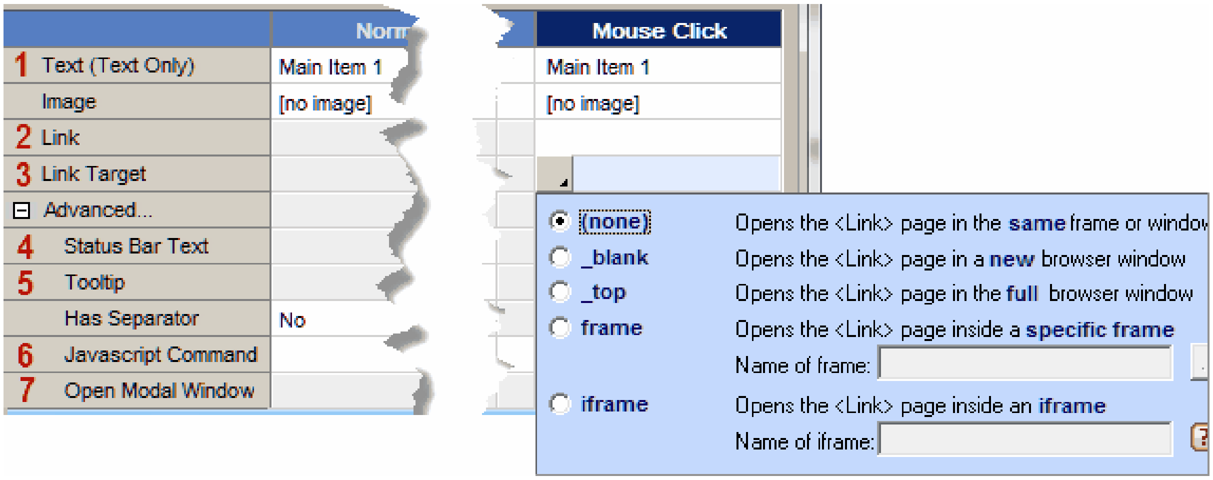
Item Properties ID: it0

	Normal		Mouse Over		Mouse Click
1 Text (Text Only)	Main Item 1		Main Item 1		Main Item 1
Image	[no image]		[no image]		[no image]
2 Link					
3 Link Target					
⊟ Advanced...					
4 Status Bar Text			%Link%		
5 Tooltip					
Has Separator	No				
6 Javascript Command					
7 Open Modal Window					

Just add/edit that cell with any link you like. More info on the "Link" property.

"Link Target" column

The cell value of this column refers to the "Link Target" AllWebMenus value (**"3"** in screenshot below).



	Norm	Mouse Click
1 Text (Text Only)	Main Item 1	Main Item 1
Image	[no image]	[no image]
2 Link		
3 Link Target		
⊟ Advanced...		
4 Status Bar Text		
5 Tooltip		
Has Separator	No	
6 Javascript Command		
7 Open Modal Window		

● (none) Opens the <Link> page in the **same** frame or windov
○ _blank Opens the <Link> page in a **new** browser window
○ _top Opens the <Link> page in the **full** browser window
○ frame Opens the <Link> page inside a **specific frame**
Name of frame:
○ iframe Opens the <Link> page inside an **iframe**
Name of iframe:

Valid values:

For the default value "*(none)*", **just leave this cell blank**. This is the most common case.

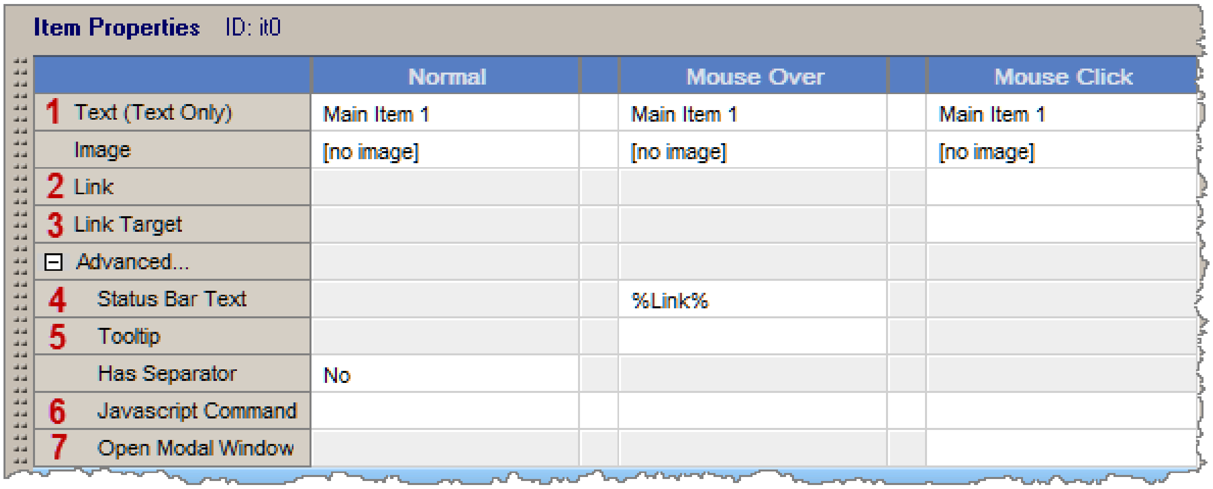
If you want a different option for an item, these are the **exact** values you are allowed to use on its "Link Target" cell:

_blank
_top
...or any name that specifies a frame/iframe in your page

More info on the "Link Target" property.

"Status Bar Text" column

The cell value of this column refers to the "Status Bar Text" AllWebMenus value (**"4"** in screenshot below).



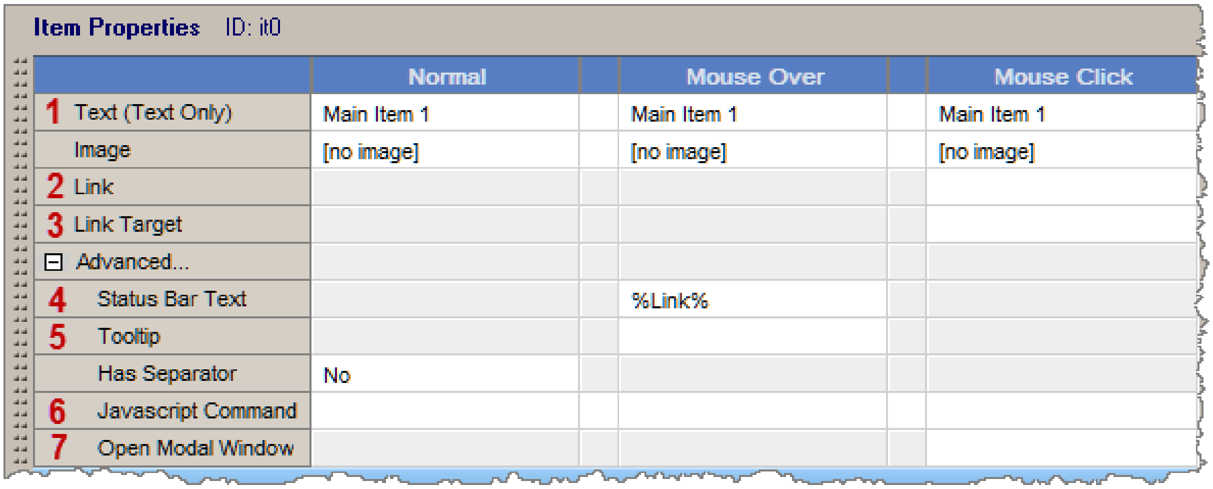
Item Properties ID: it0

	Normal	Mouse Over	Mouse Click
1 Text (Text Only)	Main Item 1	Main Item 1	Main Item 1
Image	[no image]	[no image]	[no image]
2 Link			
3 Link Target			
⊟ Advanced...			
4 Status Bar Text		%Link%	
5 Tooltip			
Has Separator	No		
6 Javascript Command			
7 Open Modal Window			

Just add/edit that cell with any text you like. More info on the "Status Bar Text" property and what %Link% means.

"Tooltip" column

The cell value of this column refers to the "Tooltip" AllWebMenus value (**"5"** in screenshot below).



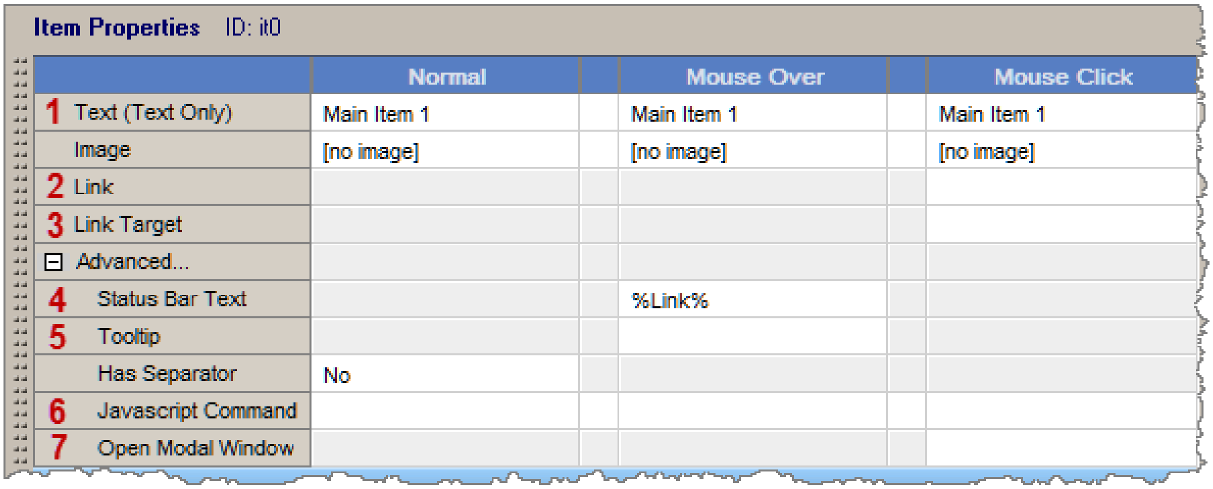
Item Properties ID: it0

	Normal	Mouse Over	Mouse Click
1 Text (Text Only)	Main Item 1	Main Item 1	Main Item 1
Image	[no image]	[no image]	[no image]
2 Link			
3 Link Target			
⊟ Advanced...			
4 Status Bar Text		%Link%	
5 Tooltip			
Has Separator	No		
6 Javascript Command			
7 Open Modal Window			

Just add/edit that cell with any text you like. More info on the "Tooltip" property.

"Javascript Command" columns (3 columns per state)

The cell values of these columns refer to the "Javascript Command" AllWebMenus values (**"6"** in screenshot below).



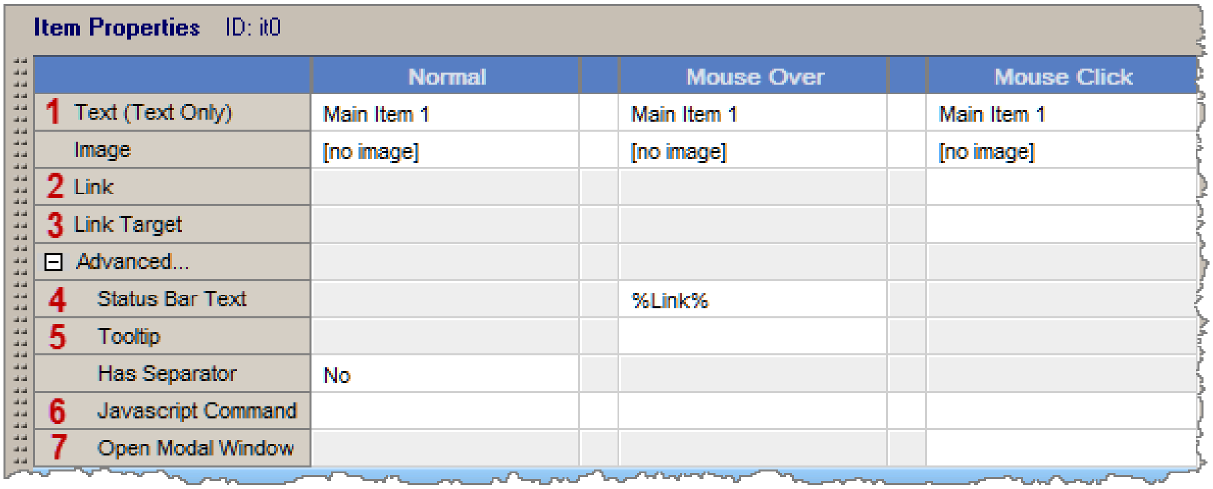
Item Properties ID: it0

	Normal		Mouse Over		Mouse Click
1 Text (Text Only)	Main Item 1		Main Item 1		Main Item 1
Image	[no image]		[no image]		[no image]
2 Link					
3 Link Target					
⊟ Advanced...					
4 Status Bar Text			%Link%		
5 Tooltip					
Has Separator	No				
6 Javascript Command					
7 Open Modal Window					

Just add/edit that cell with any text you like. More info on the "Javascript Command" property.

"Open Modal Window" column

The cell value of this column refers to the "Open Modal Window" AllWebMenus value (**"7"** in screenshot below).



Item Properties ID: it0

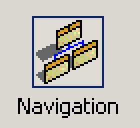
	Normal		Mouse Over		Mouse Click
1 Text (Text Only)	Main Item 1		Main Item 1		Main Item 1
Image	[no image]		[no image]		[no image]
2 Link					
3 Link Target					
⊟ Advanced...					
4 Status Bar Text			%Link%		
5 Tooltip					
Has Separator	No				
6 Javascript Command					
7 Open Modal Window					

Just add/edit that cell with any text you like. More info on the "Open Modal Window" property.

4.6.6 Command: Import FrontPage Navigation Structure

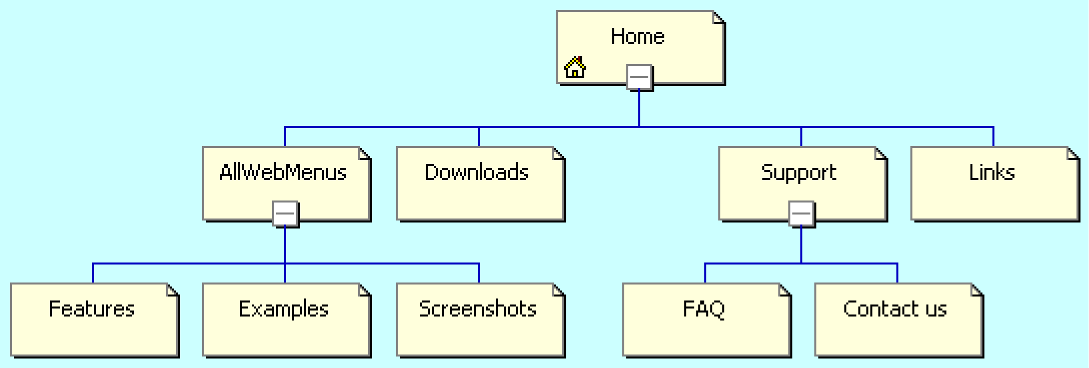
Imports the Navigation Structure of a FrontPage Web into AllWebMenus.

Note: For detailed information on how to use AllWebMenus with FrontPage, please visit http://www.likno.com/frontpage.html

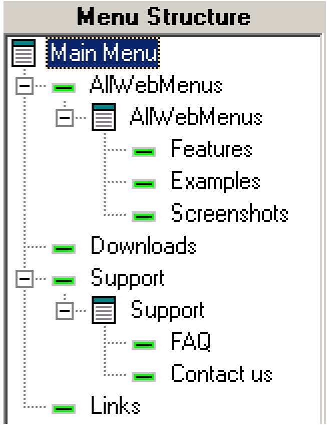


FrontPage users can create Navigation Structures inside their Webs, so that navigation buttons appear throughout their published web sites.
You can easily replace these navigation buttons with neat menus created with AllWebMenus.

A Navigation Structure inside a FrontPage Web looks like the following:



The "Import FrontPage Navigation Structure" Tool lets you import this navigation structure into AllWebMenus and create a **menu structure** out of it.



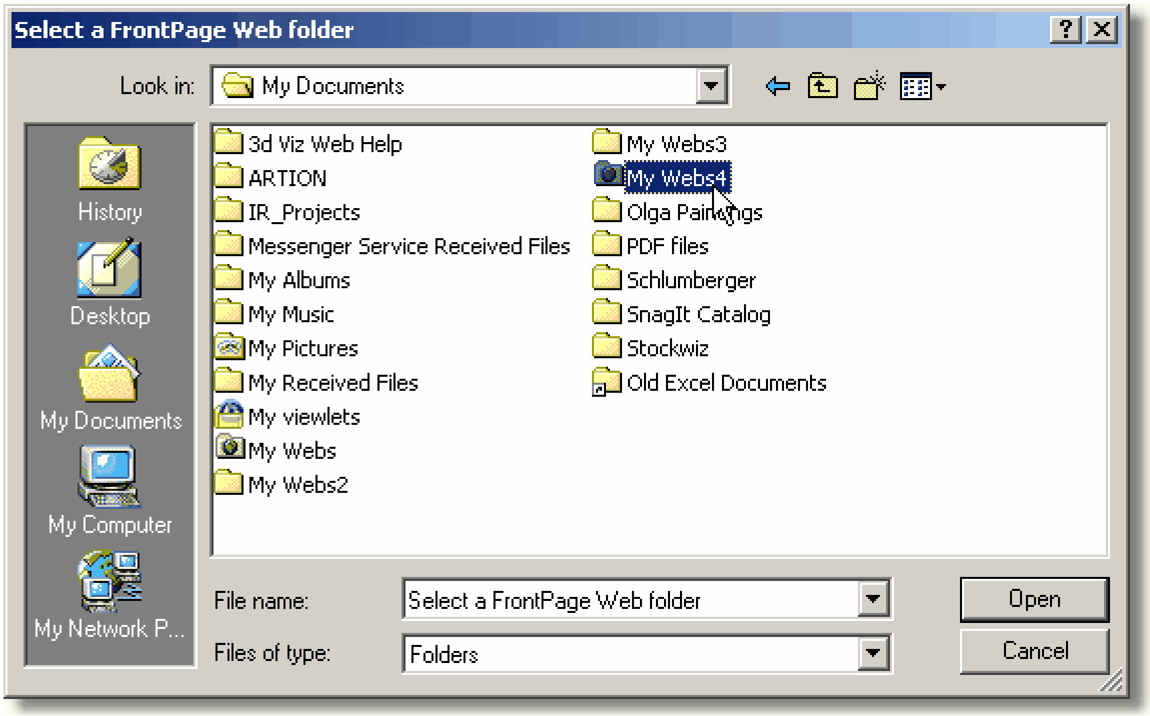
Menu Structure

- Main Menu
  - AllWebMenus
    - AllWebMenus
      - Features
      - Examples
      - Screenshots
  - Downloads
  - Support
    - Support
      - FAQ
      - Contact us
  - Links

In order to do so, you only need to select the folder where your FrontPage Web is located and AllWebMenus automatically imports the Navigation Structure.

How to select the FrontPage Web folder:

1st Step: Browse your folders until you find your particular FrontPage Web folder. Press the [Open] button.
Make sure that you selected a valid FrontPage Web folder!



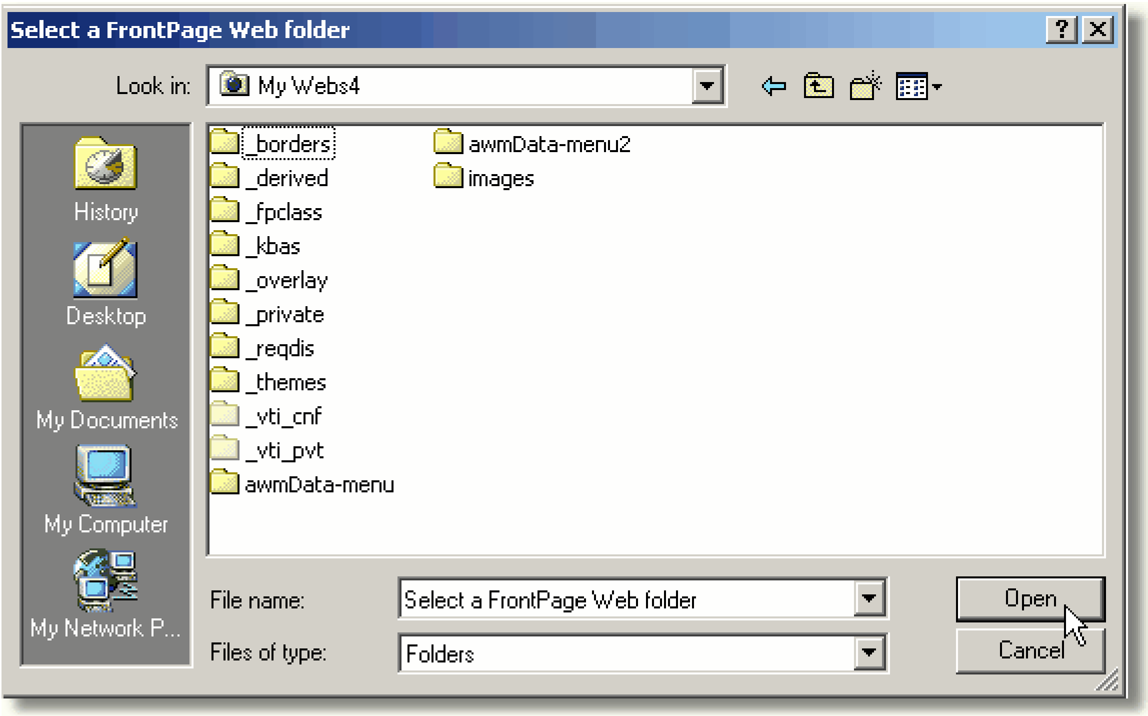
Select a FrontPage Web folder

Look in: My Documents

File name: Select a FrontPage Web folder

Files of type: Folders

2nd Step: After clicking on the FrontPage Web folder (eg: "My Webs4"), the folder opens and its contents appear.
There, you should only click the [Open] button immediately without clicking on any other folder!



Select a FrontPage Web folder

Look in: My Webs4

File name: Select a FrontPage Web folder

Files of type: Folders

You need to wait for some seconds so that AllWebMenus connects to your FrontPage installation. Provided that a FrontPage installation exists on the same computer where AllWebMenus is installed, AllWebMenus imports the Navigation Structure and creates a new menu out of it.

Note! The imported Navigation Structure uses the styles that exist in the Style Editor at the time of the import. If you want a different theme for your menu you can open the Theme Gallery and choose a different theme from there. Theme Gallery allows you to create your own themes, so it is highly recommended that you first create and add your preferred themes into the Gallery and then use them on your imported menus.

4.6.7 Create Sitemaps

4.6.7.1 Command: Create Generic Sitemap

Generates a Site Map based on your menu structure.

The Site Map is exported into a simple .html file and contains all items and their links (for those items that have links to other pages).

This is a very useful feature for two reasons:

1. It is a neat solution for **SEO** (Search Engine Optimization) purposes. It can make your website **"search-engine friendly"**, as search engine crawlers use internal links (<a> tags) to spider a site (not javascript links).

Just add a "Site Map" link at your **index** page (or any other page crawled by search engines) and your most important pages of your website (those in your menu) are seen by the search engine crawlers.

2. It can be used for **documentation, by printing the menu structure**. The Site Map (i.e. the menu structure) is saved into an .html file so you can print this .html file through your favorite browser/authoring tool.

You can select from a variety of options for the exact form of your Site Map. These include Links and Items, Items as Links, Only links or just the Items of the menu.

You can also select from a list of styles for the Site Map or of course you can create your own. To create your own style just enter your related .css file inside the /styles folder of your AllWebMenus installation.

Once the Site Map is created in simple **html format** you can use your favourite authoring tools to modify it and add the related code to your web pages.

Example:
The following site map has the same information as our menu located on our site. It was created using AllWebMenus by simply exporting its structure with the use of the "Create Site Map" Tool.

Site Menu

- Home Page
- AllWebMenus
  - Product Description
  - Examples
    - Dots Example
    - Menubar Example
    - Glass Buttons Example

        - GreenHeart Example
        - Ladder Example
        - NS-Like Toolbar Example
        - more examples...
        - Menu-driven Customer Pages!
    - Screenshots
        - ScreenShots
    - Testimonials
    - **Download** AllWebMenus
    - **Activate** AllWebMenus
    - Affiliate/Reseller Program
- Support
    - On-line Support Center
    - F.A.Q.
    - How Do I?
    - What's New!
    - E-mail Sales
- Download Area
    - Download AllWebMenus
    - Download Menus & Styles
    - Download Manual
- Interesting Links
    - Menu-driven Customer Pages!
    - HTML Links
    - Web Design Links

See also
Create Google Sitemap
Create ROR Sitemap

4.6.7.2 Command: Create Google Sitemap

Generates a Google Sitemap file (sitemap.xml) based on the linked pages of the menu items.

What is a Google Sitemap

A Google Sitemap allows you to inform search engine crawlers about URLs on your Web sites that are available for crawling. A Sitemap consists of a list of URLs and may also contain additional information about those URLs, such as when they were last modified, how frequently they change, etc.

Click here for online help from Google

How the AllWebMenus feature helps

AllWebMenus lets you create a Google Sitemap using all the links included in your menu structure (therefore, the most important links of your website).

If a menu item does not link to a page (i.e. its <Link> property is blank) it appears in gray color and will not be included in the generated Google Sitemap xml file.

You can select values for the following properties:

Last Modified Date
Change Frequency
Priority

The <default> value means that the xml file will not include any special value for the selected entry. Google will treat this entry using its default values.

The <automatic> value means that AllWebMenus reads the actual Last Modified Date of the associated files when the Sitemap generation process starts.

The <Base URL> property should be the **online root** folder of the website.

The <Location> property is the **local** root folder of your website. This is where the file **sitemap.xml** will be created. You should not forget to upload the file on your server so that Google can track and use it.

We welcome any feedback that you believe might help us improve this feature.

See also
Create Generic Sitemap
Create ROR Sitemap

4.6.7.3 Command: Create ROR Sitemap

Generates a ROR Sitemap file (ror.xml) based on the linked pages of the menu items.

What is ROR

ROR is a rapidly growing XML format for describing your website in a generic fashion, so that search engines can find and discover information more easily. ROR information is stored in a ROR file called ror.xml placed in the top directory (root folder) of your website. ROR can describe a large variety of objects: sitemaps, products, services, articles, images, menus, feeds, contact information, business information, reviews, etc.

What is a ROR Sitemap

A ROR Sitemap allows you to inform search engine crawlers about URLs on your Web sites that are available for crawling. A Sitemap consists of a list of URLs and may also contain additional information about those URLs, such as when they were last modified, how frequently they change, etc.

Click here for online help from the ROR website

How the AllWebMenus feature helps

AllWebMenus lets you create an ROR Sitemap using all the links included in your menu structure (therefore, the most important links of your website).

If a menu item does not link to a page (i.e. its <Link> property is blank) it appears in gray color and will not be included in the generated ROR Sitemap xml file.

You can select values for the following properties:

Last Modified Date

Change Frequency
Priority

The <default> value means that the xml file will not include any special value for the selected entry.

The <automatic> value means that AllWebMenus reads the actual Last Modified Date of the associated files when the Sitemap generation process starts.

The <Base URL> property should be the **online root** folder of the website.

The <Location> property is the **local** root folder of your website. This is where the file **ror.xml** will be created. You should not forget to upload this file on your server so that sites can track and use it.

We welcome any feedback that you believe might help us improve this feature.

See also
Create Generic Sitemap
Create Google Sitemap

4.6.8 Command: Batch Compile Multiple Projects

Allows you to create a "list" (or more lists) of selected projects of yours, so that you re-compile them in a batch operation when you want to update them all at once.

This command is very convenient to professionals that have created many menu projects for several websites.

You can collect all your projects into a "list" file, click the "Compile All" button and sit back while AllWebMenus automatically compiles all your existing projects in a "silent mode".

4.6.9 Command: Customize

Allows you to customize AllWebMenus on the following settings:

General

Defaults

- Preview Panel Settings
- Project Info Settings
- RTL Languages Support

4.6.9.1 General

Here you can customize the general look and feel of your AllWebMenus interface.

Show text on toolbar buttons

You can choose not to show text on the toolbar buttons (default behavior) so that you save some space for your design panels.

Start with expanded "Customize View"

You can choose to always start with the "Customize View" expanded so that you see all the style properties on the screen. The default behavior is to hide the complexity of style properties be keeping them closed.

Open Welcome Dialog on Startup

This is where you choose if the Welcome Dialog window opens on startup or not.
By default, this window opens on startup.

Launch Theme Gallery and Use Sample Items on New Menus

This is where you choose how a **new menu** appears, when selecting the "**File->New**" command:

Checked option (default):

Selecting "File->New", the Style Gallery is launched so that you can choose a style for your new menu and the new menu appears with the following structure:

```
Main Menu
  Item1
  Item2
    Submenu4
        Item5
        Item6
        Item7
  Item3
```

The checked value is the default, aiming to show you how the Style Gallery and the Menu Structure Pane work.

Unchecked option (default):

Selecting "File->New", no Style Gallery is launched and the new menu appears with the following simple structure:

Main Menu

This means that no sample items are added to your new menu. You can start populating your menu structure from scratch, without the need to delete or update any sample items. You can use this option when you are familiar with how the menu structure works.

4.6.9.2 Defaults

Allows you to customize your AllWebMenus preferences on the following properties and options:

Preview Panel Settings

Project Info Settings

RTL Languages Support

Note that the above settings refer to the initial values (defaults) that new menu projects should start with. If you want to update the values of your loaded project you should edit its Project Properties instead.

4.6.9.2.1 Preview Panel Settings

Here you can specify **default** settings for **new** menus concerning their Preview Panel.

For more info on these settings, please refer to the related project properties.

4.6.9.2.2 Project Info Settings

Here you can specify **default** settings for **new** menus concerning their Project Info.

For more info on these settings, please refer to the related project properties.

4.6.9.2.3 RTL Languages Support

Here you can specify **default** settings for **new** menus concerning their RTL Languages support (eg: Arabic, Hebrew, Persian, etc.).

For more info on these settings, please refer to the related project properties.

4.6.9.2.4 Menu-Linking Code in Tags

Here you can specify **default** settings for **new** menus concerning the "Menu-Linking Code in Tags" feature.

For more info on this feature, please refer to the related project properties.

Command: Add-ins

Part V

5 Command: Add-ins

NEW to version 4.2 and above (PRO only)

Use this section to extend the AllWebMenus capabilities by using separately bought (or free) Add-ins.

After purchasing the add-in, follow the instructions to install it. To check if this has been done properly you should be able to see its name listed in the "Add-ins" section.

To see all available add-ins select "Add-ins -> View available Add-ins online" or click directly on the URL below:

http://www.likno.com/addins/dhtml-menu-addins.html?source=app

Command: Help

Part VI

6 Command: Help

Contents
Displays the contents of the help file. It also supports both keyword and full-text search.

Search for Help On...
Performs a keyword search on the help file.

How do I...
Gives advice on how to design menus and how to do particular tasks.

Tip of the Day
Shows a tip every time you start the AllWebMenus application. You can deactivate this feature by unchecking the "Show Tips at Startup" box.

Optimization Techniques
Gives advice on how to optimize the menu design to improve the menu-building speed.

How to Purchase/Activate
Displays information about the purchase and activation process.

Purchase AllWebMenus
Transfers you to the online order form.

Enter Activation Code
This is where you can unlock your copy by typing an Activation Code. You will receive this Activation Code through your Likno User Area account which is created after your purchase of AllWebMenus.

Visit Likno on the Web
Visit our web site to contact us, get the latest information on design techniques and download more examples and styles.

Check for Updates
Checks if newer versions have been released since the last time you installed AllWebMenus.
It is recommended that you check for updates frequently as we regularly release improved cross-browser support for your menu implementations.

On-line Support Center
Opens the on-line support center page where you can find additional help information on certain features of AllWebMenus.

About
Displays the About box of AllWebMenus, with your license information.

Properties

Part VII

7 Properties

7.1 Properties Overview

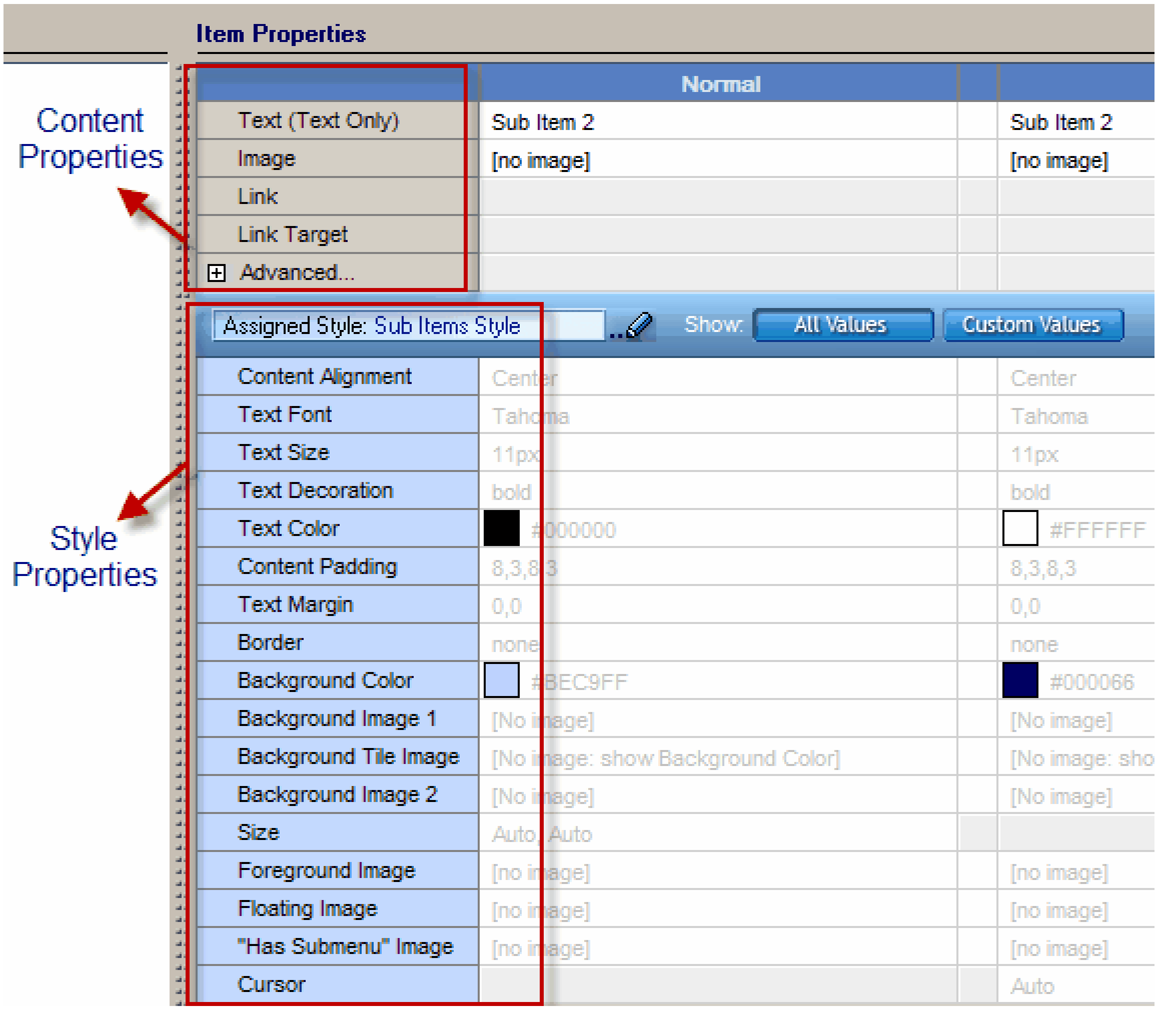
Properties are classified as follows:

According to the **element** (group or item) they describe:

- Group Properties
- Item Properties

According to the **type** of the feature they describe:

- Content Properties
- Style Properties



	Normal	
Text (Text Only)	Sub Item 2	Sub Item 2
Image	[no image]	[no image]
Link		
Link Target		
⊞ Advanced...		

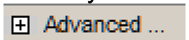
Assigned Style: Sub Items Style ... Show: All Values Custom Values

Content Alignment	Center	Center
Text Font	Tahoma	Tahoma
Text Size	11px	11px
Text Decoration	bold	bold
Text Color	■ #000000	□ #FFFFFF
Content Padding	8,3,8,3	8,3,8,3
Text Margin	0,0	0,0
Border	none	none
Background Color	□ #BEC9FF	■ #000066
Background Image 1	[No image]	[No image]
Background Tile Image	[No image: show Background Color]	[No image: sho
Background Image 2	[No image]	[No image]
Size	Auto, Auto	
Foreground Image	[no image]	[no image]
Floating Image	[no image]	[no image]
"Has Submenu" Image	[no image]	[no image]
Cursor		Auto

7.2 Property groups

7.2.1 Group Properties

Group Properties deal with the behavior and appearance of the area that contains a group of items. Groups are the Main Menu and all the Submenus.
What is the difference between the Main Menu Group and the Submenu Group?

Group Properties	Version
Group Content Properties *(properties that describe the actual content of the group. Content properties do **not** change when a new Theme is applied or when a new Style is assigned to the group):*	
Header Text	Lite & Pro
Header Image	Pro only
Positioning (Main Menu only)	Lite & Pro
Movable (Main Menu only)	Pro only
Menu Floats on Scroll (Main Menu only)	Pro only
Sticky Submenus	Pro only
Show Item or Group on Appear	Pro only
Advanced section properties:	
Header Tooltip	Pro only
Footer Text	Pro only
Footer Image	Pro only
Footer Tooltip	Pro only
Z-Index (Main Menu only)	Lite & Pro
Javascript Command on Appear	Pro only
Javascript Command on Disappear	Pro only
Group Style Properties *(properties that describe the appearance and behavior of the group, rather than its actual content. Style properties change when a new Theme is applied or when a new Style is assigned to the group):*	
Assigned Style	Lite & Pro
Type	Lite & Pro
Has Header	Lite & Pro
Has Footer	Pro only
Appears - Where (submenus only)	Lite & Pro
Appears - Aligned with (submenus only)	Pro only
Appears - How	Pro only
Disappears - How	Pro only
Opacity	Pro only
Background Color	Lite & Pro
Background Tile Image	Pro only
Padding	Pro only
Border	Lite & Pro
Rounded Corners	Pro only
Distance Between Items	Lite & Pro
Width	Pro only

For help on a specific property click on its related link.

See also
Properties Overview

7.2.2 Item Properties

Item Properties deal with the behavior and appearance of an individual item. An item property has three different values, one for each item state.

Item Properties	Version

***Item Content Properties** (properties that describe the actual content of the item. Content properties do **not** change when a new Theme is applied or when a new Style is assigned to the item):*

Text	Lite & Pro
Image	Pro only
Link	Lite & Pro
Link Target	Pro only
Status Bar Text	Pro only
Tooltip	Pro only
Has Separator	Pro only
Javascript Command	Pro only
Open Modal Window	Pro only

***Item Style Properties** (properties that describe the appearance and behavior of the item, rather than its actual content. Style properties change when a new Theme is applied or when a new Style is assigned to the item):*

Assigned Style	Lite & Pro
Content Alignment	Lite & Pro
Text Font	Lite & Pro
Text Size	Lite & Pro
Text Decoration	Lite & Pro
Text Color	Lite & Pro
Padding	Lite & Pro
Text Margin	Lite & Pro
Border	Lite & Pro
Rounded Corners	Pro only
Background Color	Lite & Pro
Background Image 1	Pro only
Background Tile Image	Pro only
Background Image 2	Pro only
Size	Pro only
Foreground Image	Pro only
Floating Image	Pro only
"Has Submenu" Image	Pro only
Cursor	Pro only

For help on a specific property click on its related link.

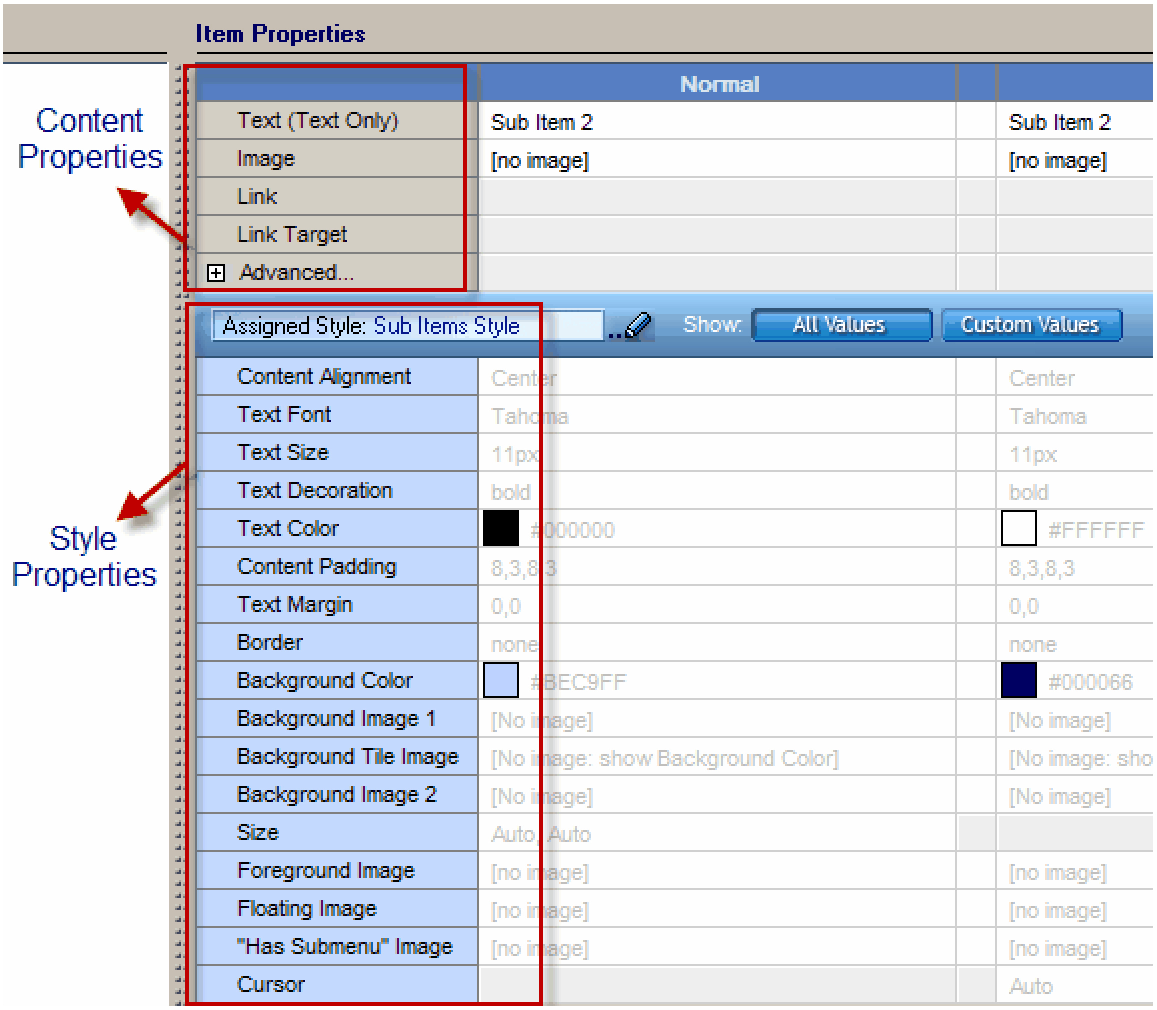
See also
Properties Overview

The Three Item states

7.3 Property types

7.3.1 Style Properties

Style Properties are used to specify the element's look and behavior, rather than its actual content. Style properties change when a new Theme is applied or when a new Style is assigned to the element (group or item).



Item Properties

	Normal		
Text (Text Only)	Sub Item 2		Sub Item 2
Image	[no image]		[no image]
Link			
Link Target			
⊞ Advanced...			

Assigned Style: Sub Items Style ... Show: All Values Custom Values

Content Alignment	Center		Center
Text Font	Tahoma		Tahoma
Text Size	11px		11px
Text Decoration	bold		bold
Text Color	■ #000000		□ #FFFFFF
Content Padding	8,3,8,3		8,3,8,3
Text Margin	0,0		0,0
Border	none		none
Background Color	□ #BEC9FF		■ #000066
Background Image 1	[No image]		[No image]
Background Tile Image	[No image: show Background Color]		[No image: sho
Background Image 2	[No image]		[No image]
Size	Auto, Auto		
Foreground Image	[no image]		[no image]
Floating Image	[no image]		[no image]
"Has Submenu" Image	[no image]		[no image]
Cursor			Auto

An item or group will use all style properties contained in its Assigned Style (except from any customized style properties that the item or group may have).

In addition, you may use the Paste Values feature to copy and paste all or some of the Style Properties to similar elements (groups or items).

The following list shows all Style properties:

Group Style Properties	*Version*
Assigned Style	Lite & Pro
Type	Lite & Pro
Has Header	Lite & Pro
Has Footer	Pro only
Appears - Where (submenus only)	Lite & Pro
Appears - Aligned with (submenus only)	Pro only
Appears - How	Pro only
Disappears - How	Pro only
Opacity	Pro only
Background Color	Lite & Pro
Background Tile Image	Pro only
Padding	Pro only
Border	Lite & Pro
Rounded Corners	Pro only
Distance Between Items	Lite & Pro
Width	Pro only

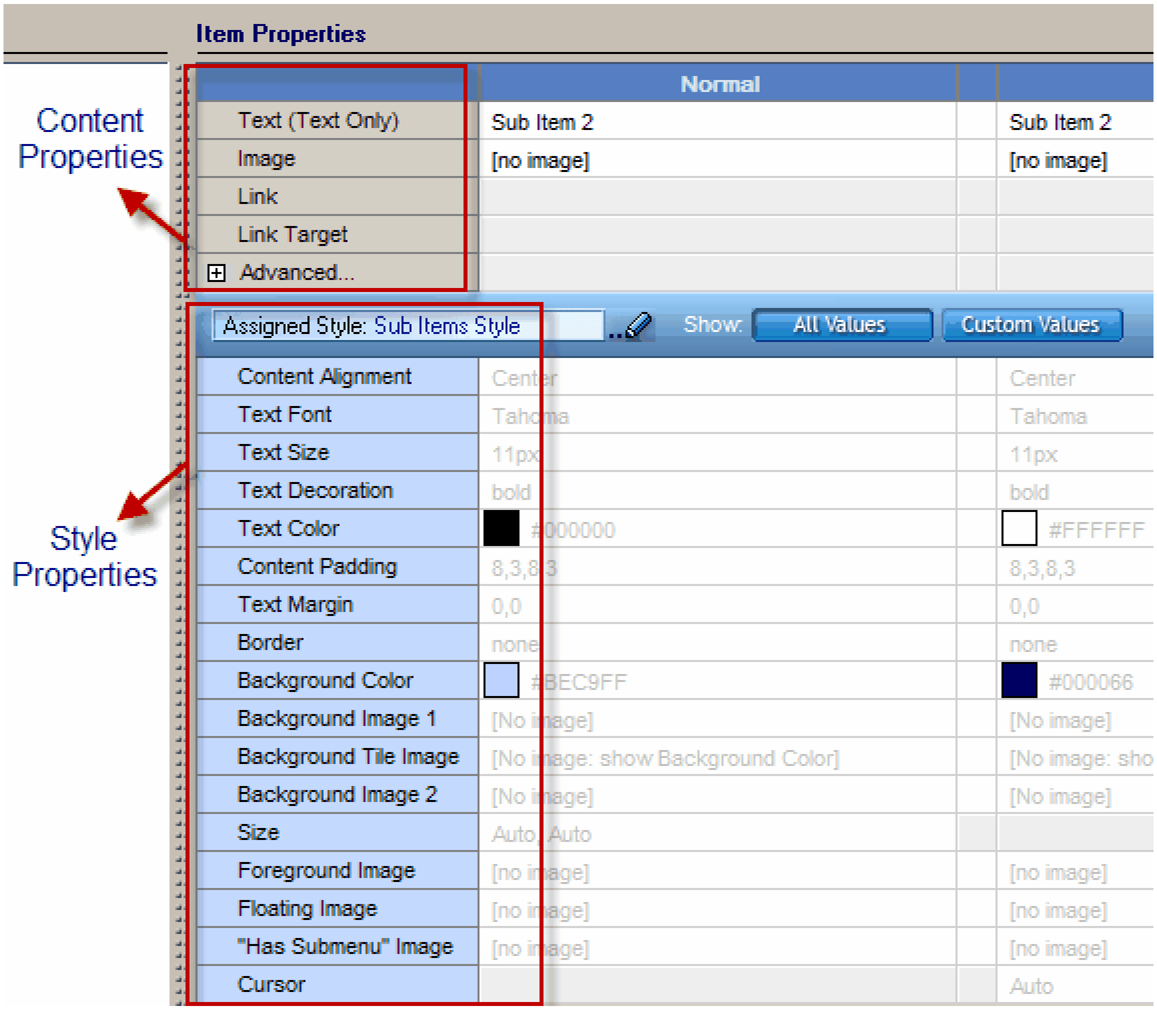
Item Style Properties	*Version*
Assigned Style	Lite & Pro
Content Alignment	Lite & Pro
Text Font	Lite & Pro
Text Size	Lite & Pro
Text Decoration	Lite & Pro
Text Color	Lite & Pro
Padding	Lite & Pro
Text Margin	Lite & Pro
Border	Lite & Pro
Rounded Corners	Pro only
Background Color	Lite & Pro
Background Image 1	Pro only
Background Tile Image	Pro only
Background Image 2	Pro only
Size	Pro only
Foreground Image	Pro only
Floating Image	Pro only
"Has Submenu" Image	Pro only
Cursor	Pro only

For help on a specific property click on its related link.

See also
Content Properties
Paste Values

7.3.2 Content Properties

Content Properties are used to specify the element's content, such as its text, link, tooltip etc. Content properties are different from the Style properties in that they do not change when a new Theme is applied or when a new Style is assigned to the element (group or item).



	Normal		
Text (Text Only)	Sub Item 2		Sub Item 2
Image	[no image]		[no image]
Link			
Link Target			
⊞ Advanced...			

Assigned Style: Sub Items Style ... Show: All Values Custom Values

Content Alignment	Center		Center
Text Font	Tahoma		Tahoma
Text Size	11px		11px
Text Decoration	bold		bold
Text Color	■ #000000		□ #FFFFFF
Content Padding	8,3,8,3		8,3,8,3
Text Margin	0,0		0,0
Border	none		none
Background Color	□ #BEC9FF		■ #000066
Background Image 1	[No image]		[No image]
Background Tile Image	[No image: show Background Color]		[No image: sho
Background Image 2	[No image]		[No image]
Size	Auto, Auto		
Foreground Image	[no image]		[no image]
Floating Image	[no image]		[no image]
"Has Submenu" Image	[no image]		[no image]
Cursor			Auto

The following list shows all Content properties:

Group Content Properties	***Version***
Header Text	Lite & Pro
Header Image	Pro only
Positioning (Main Menu only)	Lite & Pro
Movable (Main Menu only)	Pro only
Menu Floats on Scroll (Main Menu only)	Pro only
Sticky Submenus	Pro only
Show Item or Group on Appear	Pro only
Advanced section properties:	⊞ Advanced ...
Header Tooltip	Pro only
Footer Text	Pro only
Footer Image	Pro only
Footer Tooltip	Pro only
Z-Index (Main Menu only)	Lite & Pro
Javascript Command on Appear	Pro only
Javascript Command on Disappear	Pro only

Item Content Properties	***Version***
Text	Lite & Pro
Image	Pro only
Link	Lite & Pro
Link Target	Pro only
Status Bar Text	Pro only
Tooltip	Pro only
Has Separator	Pro only
Javascript Command	Pro only
Open Modal Window	Pro only

For help on a specific property click on its related link.

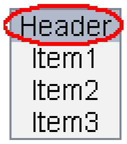
See also
Style Properties
Paste Values

7.4 Properties Description

7.4.1 Property: Header Text

Specifies the text shown on the Group's Header **if** the <Has Header> property is set to "**Yes, Show Header**".
It can also be combined with the Header Image property.

This is an example of a Header on a menu:



The <Header Text> uses the "**HTML - Formatting Mode**" only. For more info on this mode please read the related section at the Item Text property.

The <Header Text> has two uses:
- Identifies the group in the Menu Structure Pane at the left side.
- Appears on the Header when the <*Has Header*> style property is set to "Yes, Show Header". (version PRO only - Header is not supported in LITE version)

If you do not want to display a Header, you can do so by setting the <*Has Header*> style property to "No, Do not show Header".

Although blank names are not rejected, it is useful to name all elements so that they can be easily identified in the Menu Structure Pane.

Values:	Text	(HTML tags are also accepted)
	Unicode text fully supported (all languages and symbols)	
Example1:	Site Menu	
Example2:	<b>More</b> Info	

Example3: More <br> Info (very useful to break a long item name into more lines)

Applies to: Group

Type: Content Property

See also
Has Header
Header Image
Header Tooltip
Footer Text
Custom Variables

7.4.2 Property: Header Image

Displays an image on the Header of a Group.

<Header Image> appears on the Header only when the <*Has Header*> style property is set to "Yes, Show Header".

If you do not want to display a Header, you can do so by setting the <*Has Header*> style property to "No, Do not show Header".

<Header Image> can also be combined with the <Header Text>. To specify its placement related to the <Header Text> you can use the "Image Placement" option. Use "Image only" if you do not want the <Header Text> to appear next to the image.

Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values: An image file of the .gif, .jpg or .png type

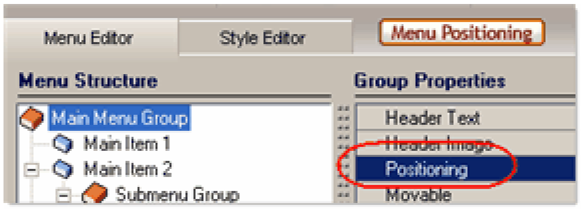
Applies to: Group

Type: Content Property

See also
Has Header
Header Text
Header Tooltip
Footer Image

7.4.3 Property: Positioning

To specify the positioning of your menu, edit the <Positioning> property located at the Main Menu group.



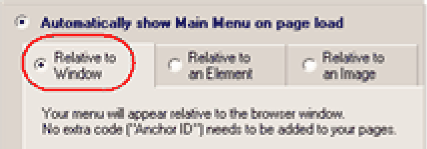
There are two methods to position your menu on your pages:

Method 1: Automatically show Main Menu on page load

This method is the default and most frequently used method for positioning your menu (i.e. the Main Menu group).

You can either position your menu relative to the browser's **window** or relative to any page **element** (div, span, table, etc.) or relative to an **image** existent on page.

Option 1: Attaching the menu to the browser's **window**

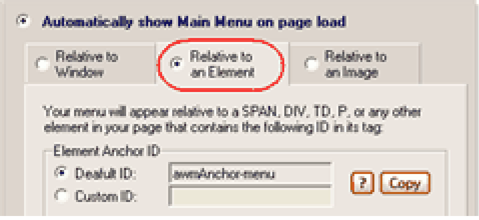

● Automatically show Main Menu on page load

● Relative to Window ○ Relative to an Element ○ Relative to an Image

Your menu will appear relative to the browser window.
No extra code ("Anchor ID") needs to be added to your pages.

This is the easiest positioning method as no extra code is required on your pages (apart from the required menu-linking code). It is an efficient solution when you want to position your menu relative to the Top-Left corner of your browser (most typical case on left-aligned websites), as this particular corner is not affected by window resizes or scrolls and therefore the menu stays fixed at the specified offsets. All you have to do is specify the Offsets X and Y and the menu will appear at that distance from your selected Window corner.

Option 2: Attaching the menu to an **element** of your pages (span, div, td, table, etc.)

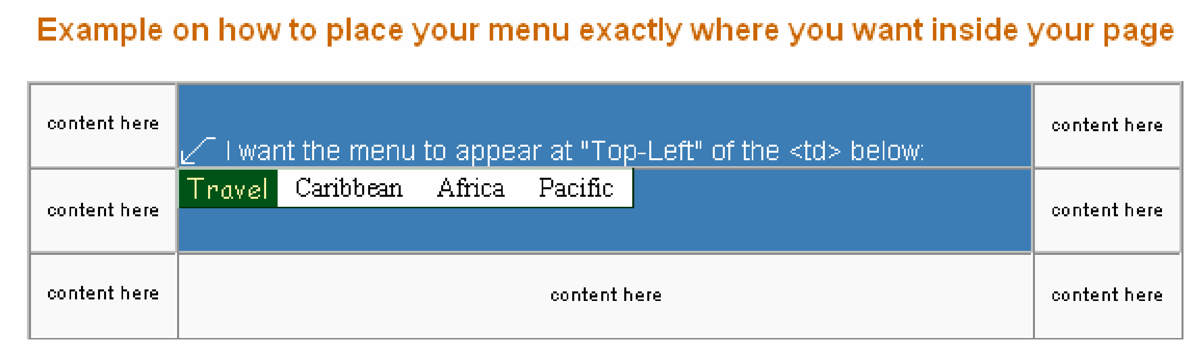

● Automatically show Main Menu on page load

○ Relative to Window ● Relative to an Element ○ Relative to an Image

Your menu will appear relative to a SPAN, DIV, TD, P, or any other element in your page that contains the following ID in its tag:

Element Anchor ID
● Default ID: awmAnchor-menu
○ Custom ID:

This option requires that you add extra code on page but gives much more flexibility and covers all positioning needs, as shown on the example below:



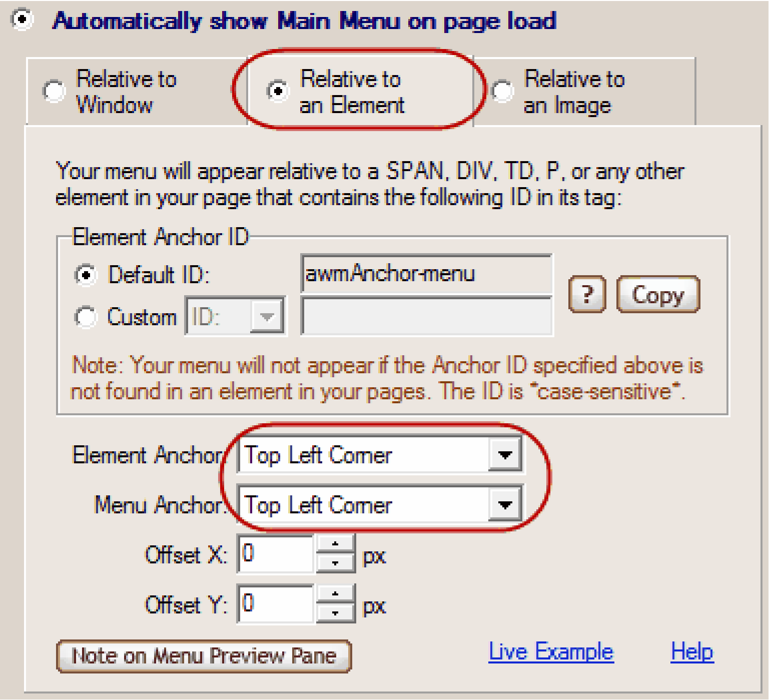
Example on how to place your menu exactly where you want inside your page

content here	I want the menu to appear at "Top-Left" of the <td> below:	content here
content here	Travel Caribbean Africa Pacific	content here
content here	content here	content here

(visit http://www.likno.com/examples/example_positioning.html to see the example)

Attaching the menu to a page element is easy to do:

1. First make sure that the "Relative to Element" option is selected at the AllWebMenus <Positioning> property before the menu compilation



● Automatically show Main Menu on page load

○ Relative to Window ● Relative to an Element ○ Relative to an Image

Your menu will appear relative to a SPAN, DIV, TD, P, or any other element in your page that contains the following ID in its tag:

Element Anchor ID

● Default ID: awmAnchor-menu

○ Custom ID:

Note: Your menu will not appear if the Anchor ID specified above is not found in an element in your pages. The ID is *case-sensitive*.

Element Anchor: Top Left Corner

Menu Anchor: Top Left Corner

Offset X: 0 px

Offset Y: 0 px

2. Edit your HTML page

3. Choose the element where you want your menu to appear

4. Add the following parameter on the element's tag:

```
id="awmAnchor-menu"
```

where ***menu*** should be the actual name of your compiled menu (without the .js extension). For example, if your compiled menu is called "*mymenu.js*" the following code should be used in your element instead:

`id="awmAnchor-`*`mymenu`*`"`

This is all you have do. When a user browses your page, the AllWebMenus script detects the element with the id parameter and shows the menu at the element's location. If the script does not detect the id, the menu will not appear.

You may also choose to use your own ID, through the ***Custom ID*** option. For example, you may be using a Content Management System that already applies its own "id" tag to every page element, thus not allowing you to add the AllWebMenus Default ID to your selected element. In that case, you simply copy this "id" of your selected element and paste it as ***Custom ID*** in AllWebMenus. Alternatively, you can use a ***Custom "Class"*** instead of an "ID", in case that your authoring tool or CMS does not allow you to edit the "ID" attribute (or the "ID" values change dynamically) but does allow you to add/edit the "Class" attribute.
For example: `<span class="menupositioner">&nbsp;</span>`

Note: IE8, Chrome and Safari vertically align the content of a table’s cell in its center, whereas all the other browsers align it on the top and that’s the reason you may observe a difference in the placement of the menu in your page. To overcome this issue, you just have to add the attribute valign="top" in the element you have positioned the menu and you will get the required functionality.

For example if you want to place your menu in a <td> element, instead of the following...

<td id=”awmAnchor”>&nbsp;</td>

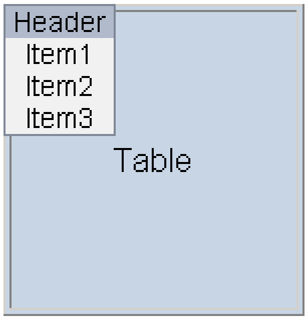
...you have to use the following:

<td id=”awmAnchor” **valign="top"**>&nbsp;</td>

Examples:

A menu (compiled as "menu.js") is attached to a table using the following code:

`<table` **`id="awmAnchor-menu"`** valign="top" `width="150" height="150" border="1">`



The same technique applies if the chosen element is different, such as:

SPAN (the most generic element):

`<span` **`id="awmAnchor-menu"`**`> &nbsp; </span>`

TD:
<td **`id="awmAnchor-menu"`** valign="top">`&nbsp;</td>`

Link:

```
<a id="awmAnchor-menu" href="#">Menu attached to this link.</a>
```

Image (you can also add the id to an image but it is recommended to use the 3rd option instead):

```
<img src="products1.gif" id="awmAnchor-menu">
```

DIV:

```
<div id="awmAnchor-menu"> &nbsp; </div>
```

Special note for positioning implemented with **AllWebMenus v4.0 and below**
(do not read if your existing menus are created and positioned with version 4.1 and above)

Prior to AllWebMenus v4.1 the following code was used for positioning:

```
<style type="text/css">.awmAnchor {position:relative;z-index:0
;left:Xpx;top:Ypx}</style>
<span id='awmAnchor-menu' class='awmAnchor'>&nbsp;</span>
```

It is ok to continue using the above code on your pages as it already contains the required Anchor ID parameter for positioning.
In case you were using OFFSETS inside the code, please bear in mind that these OFFSETS are now passed into the menu.js file instead of the above positioning code. Therefore, in order for your OFFSETS not to be *double-counted*, please make sure that you remove the following code from your html pages:

```
 <style type="text/css">.awmAnchor {position:relative;z-index:0;
left:Xpx; top:Ypx }</style>
```

We strongly recommend that you view the related "Live Example" of this property, as it shows most possible combinations on actual HTML pages.

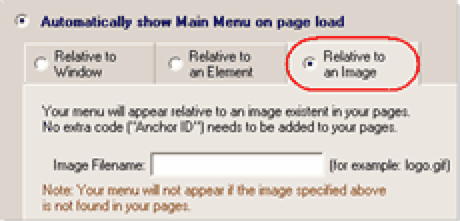
Note: Main Menu is floating - reserve space on your layout:

Remember that a DHTML menu is always **floating**. It appears **over** the page contents, not inside them. It does not push other elements away; it just **covers** them. Therefore, for the Main Menu **not to cover** other page elements you should arrange for sufficient room on your layout so that the Main Menu appears there. For example, for an horizontal menu of about 440 pixels wide and 44 pixels tall, you could use the following HTML code:

```
<table> <tr>
<td id="awmAnchor-menu" width="440" height="44">
 &nbsp;
</td>
</tr> </table>
```

You can view an on-line example about this at http://www.likno.com/examples/example_positioning.html.

Option 3: Attaching the menu to an **image** existent in your pages



● Automatically show Main Menu on page load

○ Relative to Window ○ Relative to an Element ● Relative to an Image

Your menu will appear relative to an image existent in your pages.
No extra code ("Anchor ID") needs to be added to your pages.

Image Filename: (for example: logo.gif)

Note: Your menu will not appear if the image specified above is not found in your pages.

This option is similar to the previous option ("relative to an element") but the "element" is an image that already exists in your pages (example: logo.gif) and no extra code in your pages is required (i.e. no "Anchor ID" needs to be added).

You only specify the filename of the image (which should always exist in your pages!) and your positioning is complete.

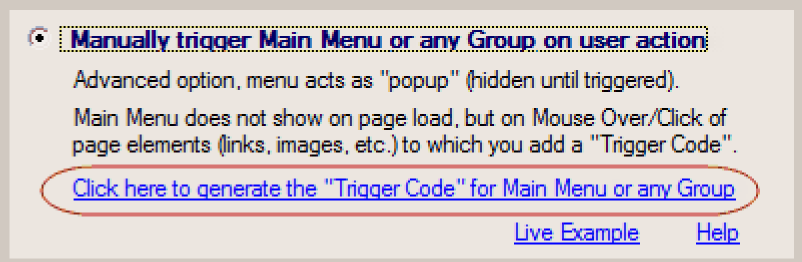
Method 2: Manually trigger Main Menu or any Group on user action (PRO version only)

With this method you have the ability to make the **Main Menu** or even **individual groups** popup by adding extra code on elements of your page. It is a very flexible method that can be used for achieving any of the following:

- Keep your entire menu hidden until an event occurs (eg: a menu popups when you put your mouse over an image or link). You can even replace the default right-click menu of your browser with your own menu!
(see "How Do I, Replace the default right-click menu used by the browser" topic)

- Let actual elements of your page (images, links, etc.) be the ones that trigger the submenus instead of Main Menu items. In this case, you have total control of the layout of your page as you do not need to keep room for a rectangle where the Main Menu appears. You can even create cyclical menus with this method! View online example

In order to get the required code that specifies which group appears and how its appearance is triggered, you need to click on the related link:



● Manually trigger Main Menu or any Group on user action

Advanced option, menu acts as "popup" (hidden until triggered).

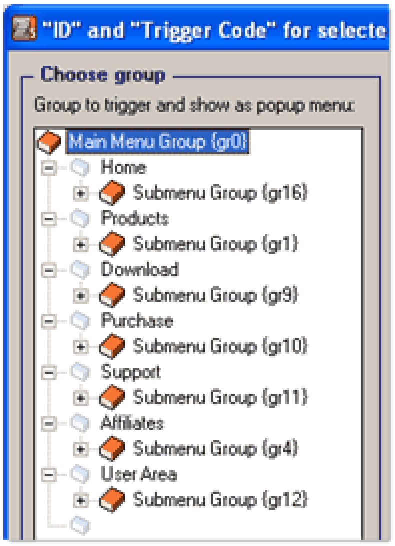
Main Menu does not show on page load, but on Mouse Over/Click of page elements (links, images, etc.) to which you add a "Trigger Code".

Click here to generate the "Trigger Code" for Main Menu or any Group

Live Example Help

This action opens a window from where you can produce the proper code to add to your pages.

Each group is assigned a unique **id** upon its creation which is enclosed in curled brackets. As shown below, the id of the Main Menu Group is **gr0**.



"ID" and "Trigger Code" for selecte

Choose group

Group to trigger and show as popup menu:

- Main Menu Group (gr0)
  - Home
    - Submenu Group (gr16)
  - Products
    - Submenu Group (gr1)
  - Download
    - Submenu Group (gr9)
  - Purchase
    - Submenu Group (gr10)
  - Support
    - Submenu Group (gr11)
  - Affiliates
    - Submenu Group (gr4)
  - User Area
    - Submenu Group (gr12)

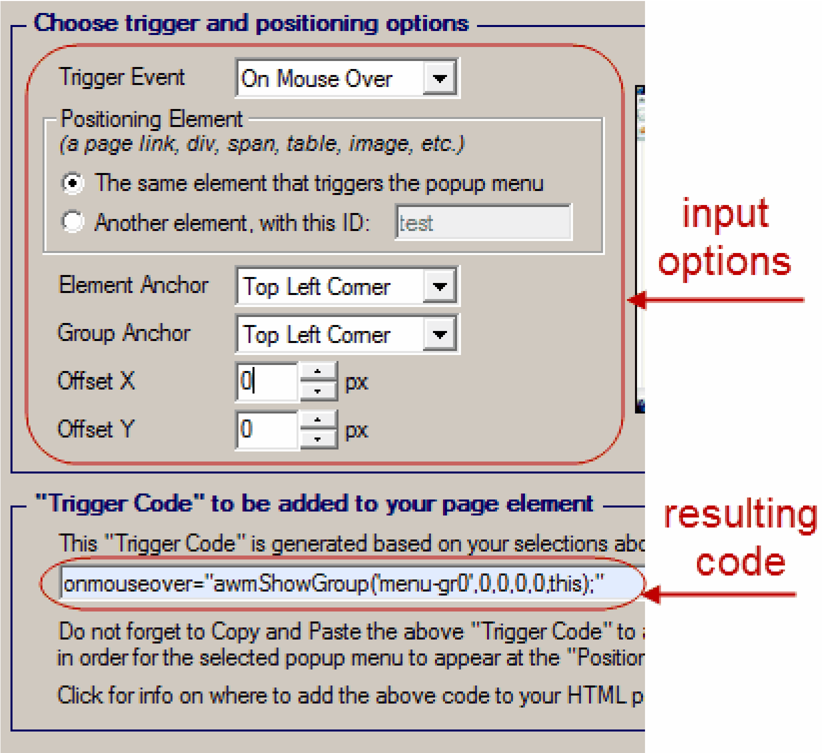
This **id** will be used within the provided code to **trigger** the appearance of the group.

For example:

```
<img src="products1.gif"
onmouseover="awmShowGroup('menu-gr2',0,3,0,0,this);">
```

In this example, we specify that the group with id **gr2** of the menu with compiled name **menu** should popup over the same element (**this**) when the user puts the **mouse over** the image.

The complete Trigger Code is generated based on the parameters that we choose on the form:



Choose trigger and positioning options

Trigger Event: On Mouse Over

Positioning Element
(a page link, div, span, table, image, etc.)

● The same element that triggers the popup menu
○ Another element, with this ID: test

Element Anchor: Top Left Corner

Group Anchor: Top Left Corner

Offset X: 0 px

Offset Y: 0 px

"Trigger Code" to be added to your page element

This "Trigger Code" is generated based on your selections abc

```
onmouseover="awmShowGroup('menu-gr0',0,0,0,0,this);"
```

Do not forget to Copy and Paste the above "Trigger Code" to a
in order for the selected popup menu to appear at the "Positior

Click for info on where to add the above code to your HTML p

Implementation Notes:

- The "Trigger Code generator" form does not have any effect on your compiled menu or page but only serves the purpose of providing the correct syntax of the code that should be added to your page. It is then your responsibility to add the provided code correctly on the page element(s) of your choice.

- As with the "Automatically show Main Menu on page load" positioning, any group can be attached to any element you like, such as images, links, table cells, divs, spans, etc.

- The "Positioning Element" (where the popup appears) can be different from the "Triggering Element". For example, a link can trigger the appearance of a popup menu over an image that is placed somewhere else in the page:

```
<p> blah blah blah... </p>
<p> <a href="#"
onclick="awmShowGroup('menu-gr0',0,0,0,0,'popup_position');">click
here!</a> </p>
<p> more blah blah blah... </p>
<p> <img id="popup_position" src="logo.gif" /> </p>
```

- *After a group appears, what makes it disappear again?"*
By default, a popup group disappears (hides) when the mouse moves in and then out from its area. To keep the popup group open even if the mouse moves away from its area you need to set the <Sticky Submenus> property to "yes". In this case, the popup group will close only when the user clicks somewhere on the page or triggers a different popup group of the same menu.
In case you want to hide an open group without entering into its area first, you can trigger the ***awmHideGroup();*** function that hides all open groups from the page, as shown on the example below:

```
<img src="products1.gif"
```

```
onmouseover="awmShowGroup('menu-gr2',0,3,0,0,this);"
onmouseout="awmHideGroup();">
```

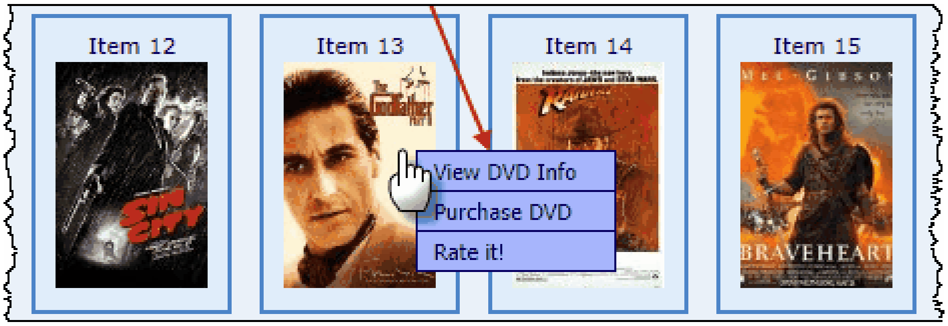
- You can use the ***awmHideMenu();*** function to hide all menus (groups) that are open on the page. The function can also contain the menu name as parameter ***awmHideMenu(menuname);*** in case you have multiple menus on page and you want to hide the open groups of a particular menu only. An example of using this function to hide a menu when passing over an image is the following:

```
<img src="products1.gif" onmouseover="awmHideMenu();">
```

- In the rare case that the AllWebMenus menu-linking code is not at the beginning of the page (as in FrontPage implementations for example), you will need to declare the ***awmShowGroup();*** function (and *awmHideMenu();* if used) in the HEAD section of your HTML page (between the <head> and </head> tags) as shown below:

```
  <script>
      function awmShowGroup(x,y,z,v,e){}
  </script>
```

- You can show the **same single menu** on different page elements, **using different <Link> values** (or "Text", "Tooltip" values) on its items per element.

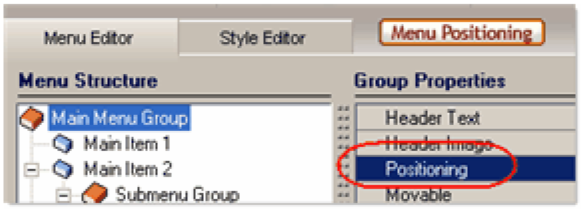


How?

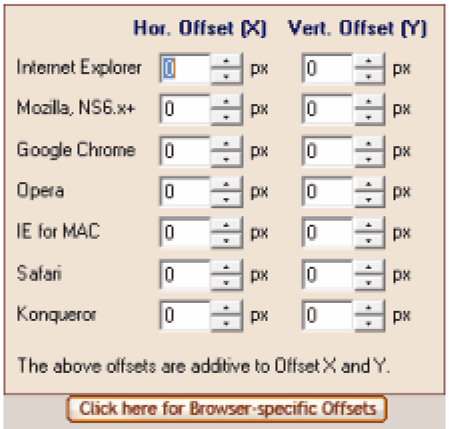
Through the Custom Variables feature (click the link to read more).

Information that relates to both Methods

Depending on your HTML structure of your pages, some browsers **may** render the menu in a slightly different position than others (some pixels off). In the **rare** situation that this occurs, you can easily fix this by using the **Browser-specific Offsets** feature, which is located at the <Positioning> property:



Note: Any browser-specific offset will be added to Offset X/Offset Y of the chosen positioning method



	Hor. Offset (X)		Vert. Offset (Y)	
Internet Explorer	0	px	0	px
Mozilla, NS6.x+	0	px	0	px
Google Chrome	0	px	0	px
Opera	0	px	0	px
IE for MAC	0	px	0	px
Safari	0	px	0	px
Konqueror	0	px	0	px

The above offsets are additive to Offset X and Y.

For each browser (or family of browsers) there are two variables that adjust the menu positioning horizontally (X) or vertically (Y) for that particular browser:

The following example shows how to use these variables:

Let's assume that the Opera browser shows your menu 5 pixels lower compared to any other browser.

In this case you need to enter the value [-5] to the "Vert. Offset (Y)" of the "Opera" option.

This variable will force the menu to appear 5 pixels higher in Opera **only**, thus fixing the problem.

Important notes!

- **Do not move the "AllWebMenus menu-linking code" from the beginning of page** (i.e. from right after the <body> tag). If you do so, you may have problems with certain browsers! Positioning is implemented using the positioning options found at the <Positioning> property, **not** by moving the menu-linking section from the beginning of the page. See Correct Placement of menu-linking code for more info on why the position of the menu-linking code is irrelevant from the actual position where the menu is specified to appear.

- Do not use extensions inside the ***id*** variable (such as ".js" or ".awm").
  correct syntax: *id=**'awmAnchor-mymenu'***
  incorrect syntax: ~~*id='awmAnchor-mymenu**.js**'*~~

- Many browsers are case-sensitive. Therefore be careful to use the correct capitalization for "*menuname*". For example if your menu is called NavBar.js, then you should have *'awmAnchor-NavBar'* inside the span element (not *'awmAnchor-navbar'*).

- If using the <SPAN> element, please note that the ***&nbsp;*** character is **not** required but recommended. You can replace the ***&nbsp;*** with any other content you may like (such as other text or an image). While it is not required, we recommend that you do not leave the <span></span> area empty, as this will cause positioning problems in OPERA, as well as in older browsers when certain code combinations are used. Please bear in mind that the menu **floats** over the span area and therefore it **covers** part or all of the content inserted there.

Applies to: Group (Main Menu only)

Type: n/a

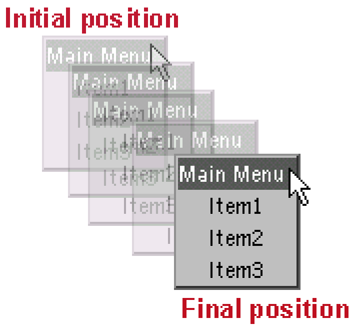
See also
Correct Placement of menu-linking code
How Do I, Replace the default right-click menu used by the browser

7.4.4 Property: Movable

Allows the viewer to relocate the menu to a different position on the web page by using the 'drag and drop' technique.

When this feature is applied the user can drag the menu from its Header (if a Header exists). To enable the Header you need to set the <Has Header> style property to "Yes, Show Header".

If the <Cursor> subproperty of the <Has Header> property is set to "Auto", then the cursor will show as "move" when positioned on the Header.



Values: yes/no

Applies to: Group (Main Menu only)

Type: n/a

7.4.5 Property: Menu Floats on Scroll

Controls the behavior of the menu when the viewer scrolls the page.

The menu can follow the scrolling direction allowing the user to view it at all times. It stays on top of the contents and moves automatically by the scrolling step. In that way, the menu retains its position on the screen regardless of the shifting of contents caused by the scrolling.

There are two options for the floating feature.

1. The "Constant Floating" option, which makes the menu "float" all the way through the page towards the scrolling direction.

2. The "Advanced Floating" option, which allows for more customization (such as floating behavior within specific sections of the page) and for different and interesting navigation cases within the page. This feature requires an extra Add-in. To see the capabilities of this additional option please view some of the available online Use Cases of the "Advanced Floating Menu Add-in".

Note: Due to technical reasons, this feature is not available on "Sliding Menu" implementations (i.e. sliding menus cannot float on scroll).

Applies to: Group (Main Menu only)

Type: n/a

7.4.6 Property: Sticky Submenus

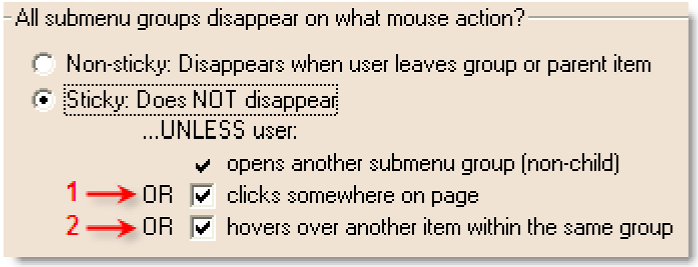
Specifies whether the submenus close when the mouse moves away from the menu area or remain open until the user clicks somewhere else on the page.

The default behavior for the submenus is to close when the mouse moves away from the menu area. If you change this property to "sticky", the submenus will only close if the user clicks somewhere on page instead.

Sticky submenus behave exactly like the submenus in the Windows "Start" menu. They close if the user clicks somewhere outside the menu area (on the page) OR hovers over another item within the same group.

Options [1] and [2] below show some new interesting variations of "sticky" submenus.

> By unchecking the default check on option [1] you can have the last opened submenu always visible, regardless of what the user does on the rest of the page.
>
> By also unchecking the default check on option [2] you can simulate a TAB-like behavior for the menu, as at least one submenu will **always** remain open no matter what happens on page or menu (see "XP Tabs" online example).



All submenu groups disappear on what mouse action?

○ Non-sticky: Disappears when user leaves group or parent item

● Sticky: Does NOT disappear

...UNLESS user:

✓ opens another submenu group (non-child)

1 → OR ☑ clicks somewhere on page

2 → OR ☑ hovers over another item within the same group

Values: Non-sticky/Sticky

Applies to: Group

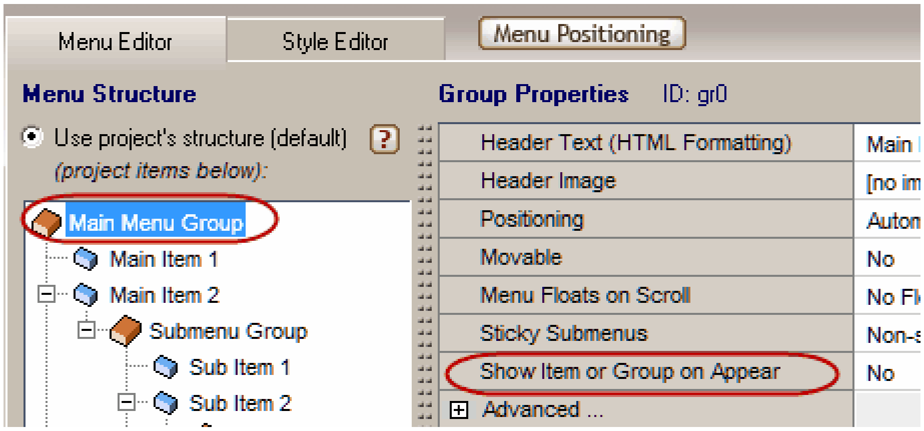
Type: n/a

7.4.7 Property: Show Item or Group on Appear

When the menu initially appears on the page, you can have an item shown as "selected" (highlighted) or a group appear open.

See an online example:
http://www.likno.com/examples.html?example=show_item_group_on_appear

This property is found in the "Main Menu Group" properties:



Menu Structure

● Use project's structure (default) ?
(project items below):

- Main Menu Group
  - Main Item 1
  - Main Item 2
    - Submenu Group
      - Sub Item 1
      - Sub Item 2

Group Properties ID: gr0

Header Text (HTML Formatting)	Main
Header Image	[no im
Positioning	Auton
Movable	No
Menu Floats on Scroll	No Fl
Sticky Submenus	Non-s
Show Item or Group on Appear	No
⊞ Advanced ...	

There are 3 options to achieve this:

Option 1 (automatically, item selected based on URL)

● Automatically, based on URL

(applies for item highlighting only)

Check this option and the menu will display as selected (highlighted) the first item whose <Link> property matches the **URL** of the loaded page *(if more than one menu item links to the same page (URL) then only the 1st matching item will be highlighted)*.

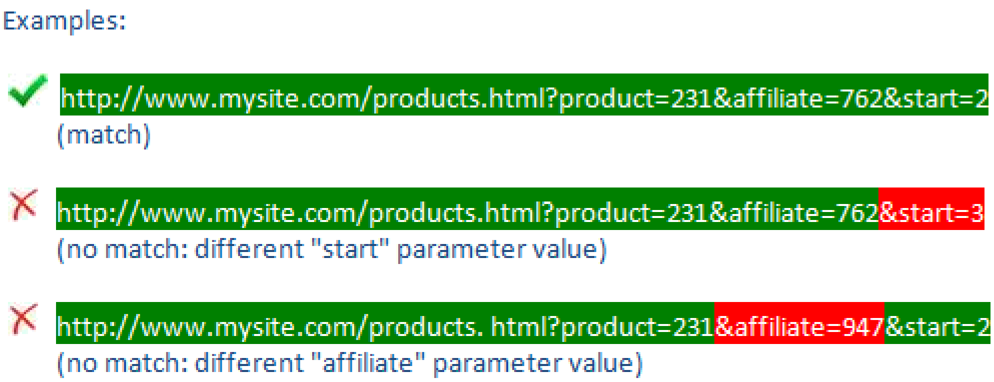
For more info on how this feature works click here.

What if a URL **uses parameters?** Should their values also be matched (or ignored)?

Example: http://www.mysite.com/products.html?product=231&affiliate=762&start=2

In this case, you have the following 3 options:

1. *Fully match* the URL (consider parameter values too)



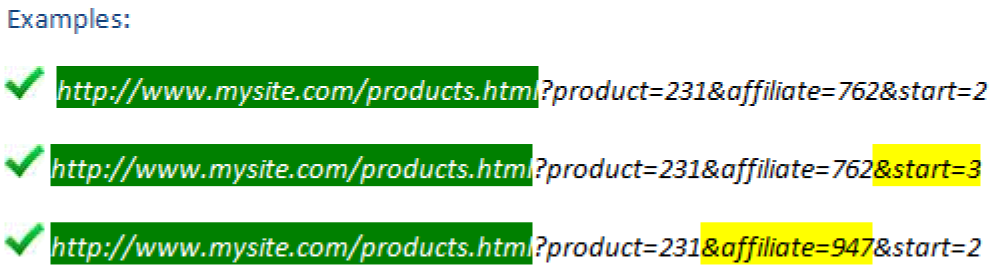
Examples:

✓ http://www.mysite.com/products.html?product=231&affiliate=762&start=2
(match)

✗ http://www.mysite.com/products.html?product=231&affiliate=762&start=3
(no match: different "start" parameter value)

✗ http://www.mysite.com/products. html?product=231&affiliate=947&start=2
(no match: different "affiliate" parameter value)

2. *Ignore all* parameter values (match only the non-parameters part of the URL)



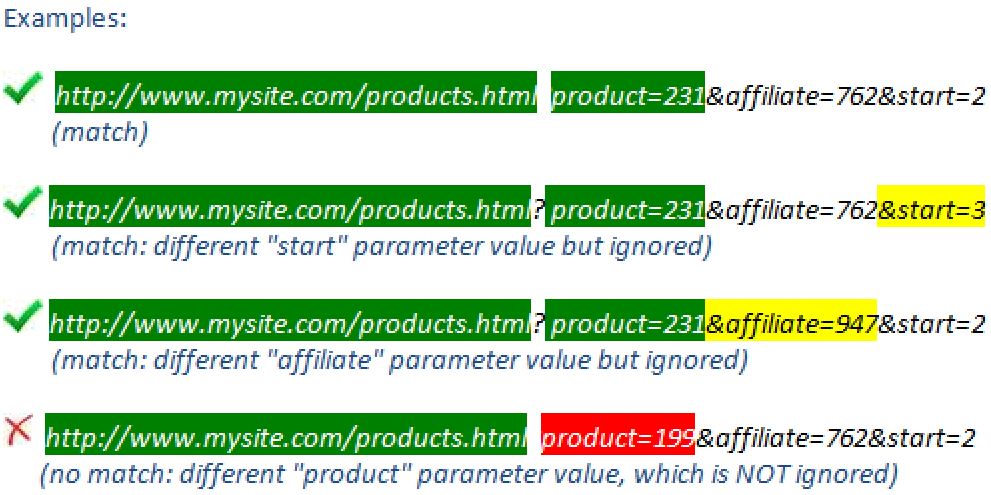
Examples:

✓ *http://www.mysite.com/products.html?product=231&affiliate=762&start=2*

✓ *http://www.mysite.com/products.html?product=231&affiliate=762&start=3*

✓ *http://www.mysite.com/products.html?product=231&affiliate=947&start=2*

3. *Ignore specific* parameters only: [affiliate, start]   (for multiple parameters use commas)



Examples:

✓ *http://www.mysite.com/products.html product=231&affiliate=762&start=2*
(match)

✓ *http://www.mysite.com/products.html? product=231&affiliate=762&start=3*
(match: different "start" parameter value but ignored)

✓ *http://www.mysite.com/products.html? product=231&affiliate=947&start=2*
(match: different "affiliate" parameter value but ignored)

✗ *http://www.mysite.com/products.html product=199&affiliate=762&start=2*
(no match: different "product" parameter value, which is NOT ignored)

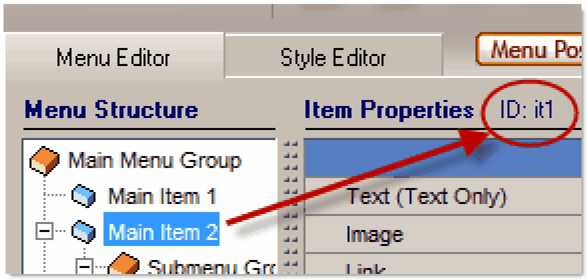
Option 2 (manually, standard item/group in all pages using the menu)

● Manually, choose a specific Item or Group:

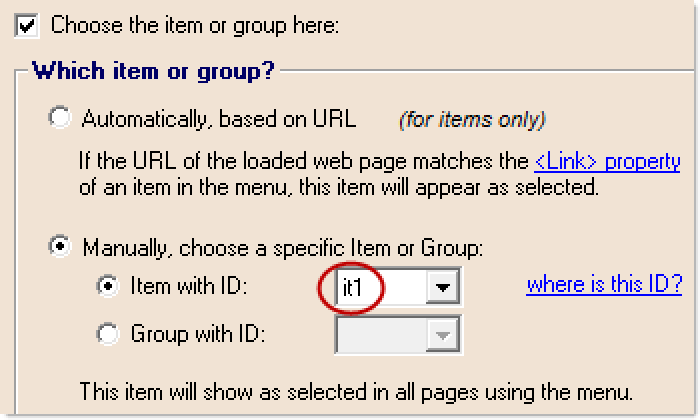
(applies for both item and group highlighting)

Check this option and the menu will display as selected (highlighted) the specific item or group that you choose from the **interface**.

The item or group is specified by its **unique ID**, which is found here:



In this example, if you want to show "Main Item 2" as selected you need to use its ID "**it1**", as demonstrated below:



☑ Choose the item or group here:

Which item or group?

○ Automatically, based on URL *(for items only)*

If the URL of the loaded web page matches the <Link> property of an item in the menu, this item will appear as selected.

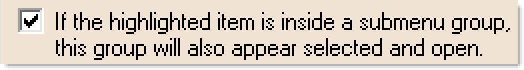
● Manually, choose a specific Item or Group:

● Item with ID: it1 where is this ID?

○ Group with ID:

This item will show as selected in all pages using the menu.

If the specified item is within a submenu group then you can choose whether its submenu will also appear selected/open or not, through this property:



☑ If the highlighted item is inside a submenu group, this group will also appear selected and open.

You may also use a group ID (instead of an item ID) if you just want to show your menu with a submenu already open (but without any of its items being selected).

Option 3 (manually, custom item/group per page through a page variable)

Use a **javascript variable** in

(applies for both item and group highlighting)

This allows you to control which item shows as selected (or which group appears open) dynamically through the HTML of your web page.

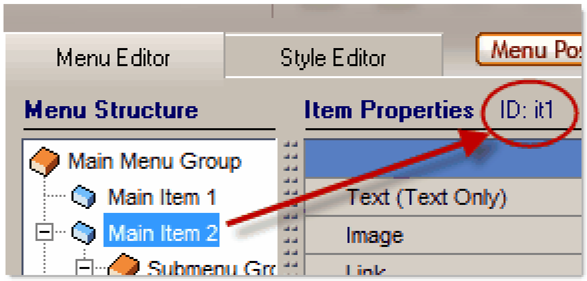
You do not have to check any option in the interface, just add the following code in a **single** line **before** the menu-linking code of your web page (the menu constructor):

```
<script type="text/javascript">var awmShowItem=["menuname,ID,
Over_Click,Show_Submenu"];</script>
```

where:

menuname is the name of the compiled file without the ".js" extension
*(example: if you compile as "menu.js" then use **menu** as the value for that parameter)*

ID is the ID of the item (or group) that will appear as initially selected, and can be found here:



*(example: **it4** OR **gr2**)*

Over_Click is a number that applies for item IDs only (not group IDs) and specifies whether the "Mouse Over" or "Mouse Click" state is used to show the item as selected. Note that this parameter is optional and the default value is 1.
It takes the value "1" or "2" (without quotes):
1 for "Mouse Over"
2 for "Mouse Click"

Show_Submenu is a number that applies for item IDs only (not group IDs) and specifies whether the submenu of the highlighted item (if it has one) should also expand. Note that this parameter is optional and the default value is 1.
It takes the value "0" or "1" (without quotes):
0 to leave the submenu closed
1 to also expand the submenu

Examples for an item:

```
<script type="text/javascript">var awmShowItem=["menu,it1,2"];
</script>
```

This means: The item with ID "it1" from the menu compiled as "menu.js" will show as selected using its "Mouse Click" state. If the item has a submenu, this will be expanded.

```
<script type="text/javascript">var awmShowItem=["menu,it2"];</script>
```

This means: The item with ID "it2" from the menu compiled as "menu.js" will show as selected using its "Mouse Over" state. If the item has a submenu, this will be expanded.

```
<script type="text/javascript">var awmShowItem=["menu,it1,2,0"];
</script>
```

This means: The item with ID "it1" from the menu compiled as "menu.js" will show as selected using its "Mouse Click" state. The submenu of this item will not be expanded.

Example for a group:

```
<script type="text/javascript">var awmShowItem=["menu,gr1"];</script>
```

This means: The group with ID "gr1" from the menu compiled as "menu.js" will show open when the menu loads.

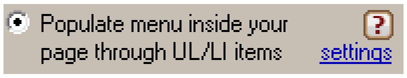
Example for selected items/groups on multiple menus!:

This feature even supports **multiple menus** on the same page, as follows:

```
<script type="text/javascript">var awmShowItem=["menu1,it4,1",
"menu2,gr3", "menu3,it8,2,0"];</script>
```

This means:
- The item with ID "it4" from the menu compiled as "menu1.js" will show as selected using its " Mouse Over" state, its submenu will also be expanded (if there is one).
- The group with ID "gr3" from the menu compiled as "menu2.js" will show open when the menu loads.
- The item with ID "it8" from the menu compiled as "menu3.js" will show as selected using its " Mouse Click" state, without expanding the submenu.

About the awmShowItem variable when the menu is populated using a "UL/LI structure".


● Populate menu inside your page through UL/LI items

In order to find the correct ID to use as a parameter, you should count the loading order of the ULs and LIs. So you should count how many LI tags are found in the UL/LI structure until the desired item is found.

For example this is a usual UL/LI code:

```
<ul id="awmULLI-menu" style="display:none;">
  <li>Main Item 1</li>
  <li>Main Item 2
    <ul>
      <li>Sub Item 1</li>
      <li>Sub Item 2
        <ul>
```

```
            <li>Sub Item 4</li>
            <li>Sub Item 5</li>
            <li>Sub Item 6</li>
          </ul>
        </li>
        <li>Sub Item 3</li>
      </ul>
    </li>
    <li>Main Item 3</li>
  </ul>
```

According to the **awmShowItem** technique, if you want to make the “Sub Item 6” highlighted, you should use **it7** since its LI tag is the 7th LI in the structure.

Notes: **-** The "Show Item or Group on Appear" feature is not available on cross-frame menu implementations (menus in framed pages).

- The previous **awmSelectedItem** variable is now deprecated by the stronger awmShowItem variable and not supported (but is still functional if it remains in your implementations). Read how it used to work (if you still use it please replace it).

- The [Mouse Over] state will be displayed instead of the [Mouse Click] state if you have the "*Ignore [Mouse Click] values*" box checked inside the Tools -> Project Properties -> Optimization tab.

- If the "Menu Floats on Scroll" property is set to "Floating" AND the "selected" item is within a submenu, then the submenu will not appear open.

Applies to: Group (Main Menu only)

Type: n/a

Keywords: 'keeping submenus open', 'always visible'

7.4.8 Property: Header Tooltip

Tooltip text that appears above the Header when the user pauses the mouse pointer over the Header.

If you do not want to display a Header, you can do so by setting the *<Has Header>* style property to "No, Do not show Header".

Values: Text
Unicode text fully supported (all languages and symbols)

Applies to: Group

Type: Content Property

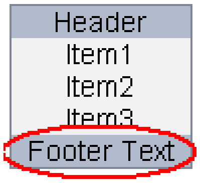
See also

Has Header
Header Text
Header Image
Footer Tooltip
Custom Variables

7.4.9 Property: Footer Text

Specifies the text shown on the Group's Footer **if** the <Has Footer> property is set to "**Yes, Show Footer**".
It can also be combined with the Footer Image property.

This is an example of a Footer on a menu:



The <Footer Text> uses the "**HTML - Formatting Mode**" only. For more info on this mode please read the related section at the Item Text property.

The <Footer Text> appears only when the <*Has Footer*> style property is set to "Yes, Show Footer". (Lite only: Footer is not supported in LITE version)

If you do not want to display a Footer, you can do so by setting the <*Has Footer*> style property to "No, Do not show Footer".

Values:	Text (HTML tags are also accepted) Unicode text fully supported (all languages and symbols)
Example1:	Site Menu
Example2:	<b>More</b> Info
Example3:	More Info (very useful to break a long item name into more lines)
Applies to:	Group
Type:	Content Property

See also
Has Footer
Footer Image
Footer Tooltip
Header Text
Custom Variables

7.4.10 Property: Footer Image

Displays an image on the Footer of a Group.

<Footer Image> appears on the Footer only when the *<Has Footer>* style property is set to "Yes, Show Footer".

If you do not want to display a Footer, you can do so by setting the *<Has Footer>* style property to "No, Do not show Footer".

<Footer Image> can also be combined with the <Footer Text>. To specify its placement related to the <Footer Text> you can use the "Image Placement" method. Use "Image only" if you do not want the <Footer Text> to appear next to the image.

Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values: An image file of the .gif, .jpg or .png type

Applies to: Group

Type: Content Property

See also
Has Footer
Footer Text
Footer Tooltip
Header Image

7.4.11 Property: Footer Tooltip

Tooltip text that appears above the Footer when the user pauses the mouse pointer over the Footer.

If you do not want to display a Footer, you can do so by setting the *<Has Footer>* style property to "No, Do not show Footer".

Values: Text

Applies to: Group

Type: Content Property

See also
Has Footer
Footer Text
Footer Image
Header Tooltip
Custom Variables

7.4.12 Property: Z-Index

In general, the *z-index* property specifies the stack order of a page element (if it shows 'in front' or 'behind' other page elements).

An element with **greater stack order (z-index value)** is always **in front** of an element with a lower *z-index* value.

In our case, the "page element" is the menu and its submenus. By default, the menu is assigned a large *z-index* value (**1000**) so that no other page element shows in front of it or its submenus.

However, in case your submenus **show behind** one of your page elements, the following may happen:

- this element is assigned a *z-index* value which is greater than the menu's value. Just lower this element's *z-index* value.

- this element is a **Flash object** (youtube video, etc.) or a **Java applet**. A higher *z-index* value **will not work** for these elements. **Special treatment** is required instead. Read instructions on how to make your submenus appear 'in front' of such elements.

Values:	from -1000 to 1000 (**0** implies no z-index assigned)
Applies to:	Group (Main Menu only)
Type:	n/a

7.4.13 Property: Keyboard Navigation

Specifies if the user can navigate through the menu by using the keyboard (tab key and arrow keys).

"Shift-F12" gives the focus to the first item of the Main Menu, so that the user can then use the tab key or arrow keys to navigate from there. When more than one menu exists in page and some of them use the "Keyboard Navigation" feature, "Shift-F12" rotates the focus among them.

"Esc" closes the current submenu (the one that has the focus).

Note: This feature cannot be tested within the "Menu Preview" pane, you will see it in your actual web page where your compiled menu is.

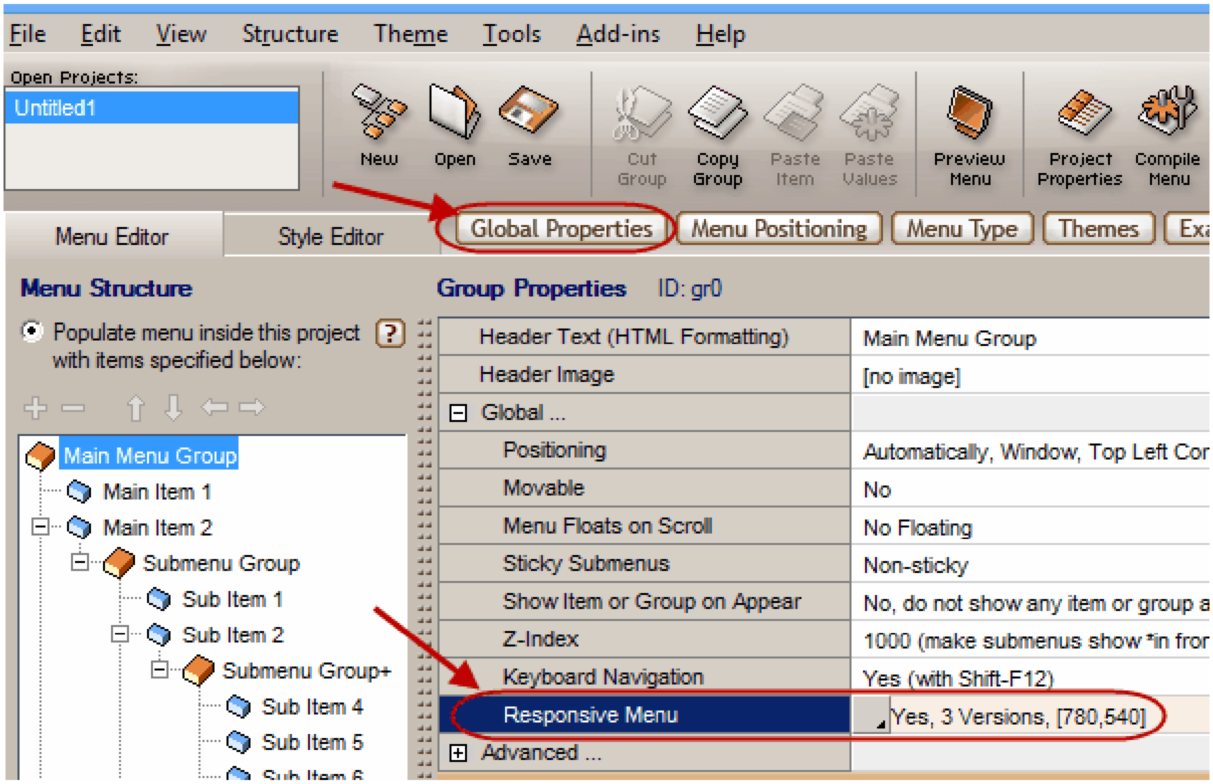
7.4.14 Property: Responsive Menu

Specifies if the menu shows a different version of it depending on screen size (width).

Each different responsive menu version of the same menu project may have different content, styling or page positioning (at any combination).

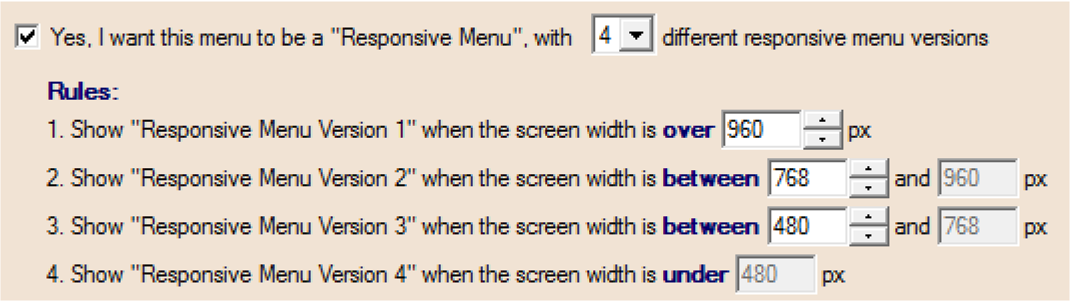
How?

Open the "Responsive Menu" property and select "Yes" along with the number of responsive menus that the loaded menu project should use.



Group Properties ID: gr0

Header Text (HTML Formatting)	Main Menu Group
Header Image	[no image]
⊟ Global ...	
Positioning	Automatically, Window, Top Left Cor
Movable	No
Menu Floats on Scroll	No Floating
Sticky Submenus	Non-sticky
Show Item or Group on Appear	No, do not show any item or group a
Z-Index	1000 (make submenus show *in fror
Keyboard Navigation	Yes (with Shift-F12)
Responsive Menu	Yes, 3 Versions, [780,540]
⊞ Advanced ...	

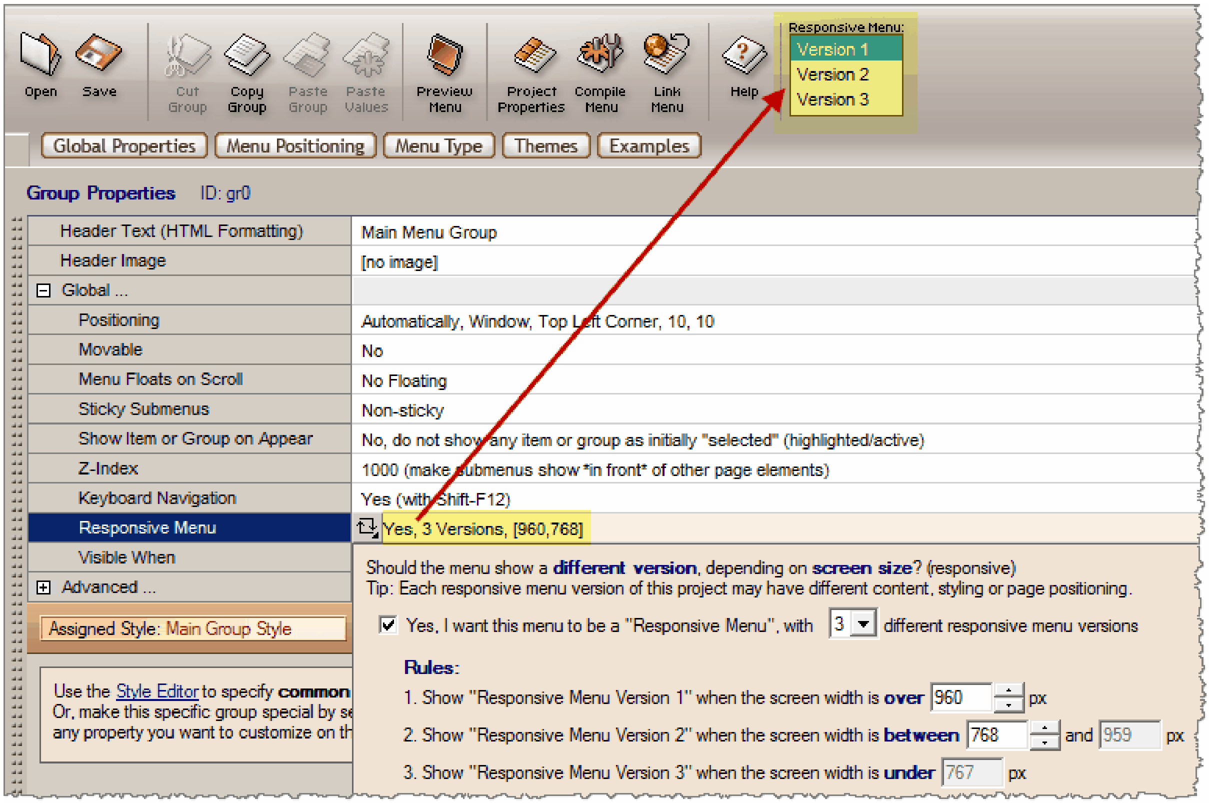


☑ Yes, I want this menu to be a "Responsive Menu", with 4 different responsive menu versions

Rules:

1. Show "Responsive Menu Version 1" when the screen width is **over** 960 px
2. Show "Responsive Menu Version 2" when the screen width is **between** 768 and 960 px
3. Show "Responsive Menu Version 3" when the screen width is **between** 480 and 768 px
4. Show "Responsive Menu Version 4" when the screen width is **under** 480 px

A listbox will then appear at the right side of the toolbar:



Group Properties ID: gr0

Header Text (HTML Formatting)	Main Menu Group
Header Image	[no image]
⊟ Global ...	
Positioning	Automatically, Window, Top Left Corner, 10, 10
Movable	No
Menu Floats on Scroll	No Floating
Sticky Submenus	Non-sticky
Show Item or Group on Appear	No, do not show any item or group as initially "selected" (highlighted/active)
Z-Index	1000 (make submenus show *in front* of other page elements)
Keyboard Navigation	Yes (with Shift-F12)
Responsive Menu	Yes, 3 Versions, [960,768]
Visible When	
⊞ Advanced ...	

Assigned Style: Main Group Style

Use the Style Editor to specify common
Or, make this specific group special by se
any property you want to customize on th

Should the menu show a **different version**, depending on **screen size**? (responsive)
Tip: Each responsive menu version of this project may have different content, styling or page positioning.

☑ Yes, I want this menu to be a "Responsive Menu", with 3 different responsive menu versions

Rules:

1. Show "Responsive Menu Version 1" when the screen width is **over** 960 px
2. Show "Responsive Menu Version 2" when the screen width is **between** 768 and 959 px
3. Show "Responsive Menu Version 3" when the screen width is **under** 767 px

Through this listbox, you can always select a different "Version" of your menu project (other than the regular "Version 1") to edit. All properties then refer to that particular version of the menu (except of some compile properties that refer to the entire project as a whole). You may then edit these properties to change the content, style or positioning values of that version. You can use the "Copy" and "Paste" functionality on the Structure or Styles among the different versions.

Note that some restrictions apply on this feature:

- You cannot have "responsive menus" when your menu project is of the CSS Menu Genre.
- You cannot have "responsive menus" when your menu project uses the "UL/LI" population method.
- You cannot have "responsive menus" when your menu project uses the "Cross-frame support" feature.

Click here to view a "Responsive Menu Example" online.

Also, read a blog post about our Responsive Menu feature.

7.4.15 Property: Visible When

Specifies when the menu shows or not, based on rules that you set.

This is useful when you want your menu to appear (or not appear) in special cases.

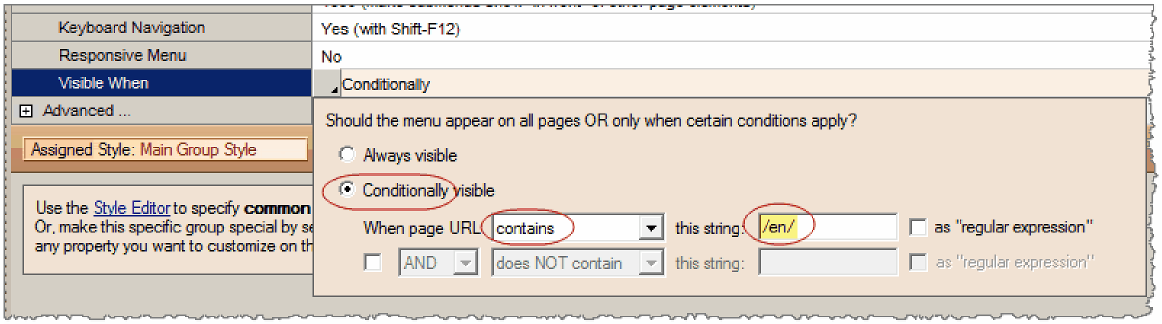
Example:

You have 2 separate menu projects, one in **English** language (menu-en.js) and one in **German** language (menu-de.js), and your website is based on a CMS through which you link **both** menus to all pages.

You want the English menu to show on the English pages of the website (eg: http://www.mysite/en/page1.html) while the German one on the respective German pages (eg: http://www.mysite/de/page1.html).

This is how you can do this:

On the **English** menu project (menu-en.js) you set the following condition:



Keyboard Navigation: Yes (with Shift-F12)
Responsive Menu: No
Visible When: Conditionally
⊞ Advanced ...
Assigned Style: Main Group Style

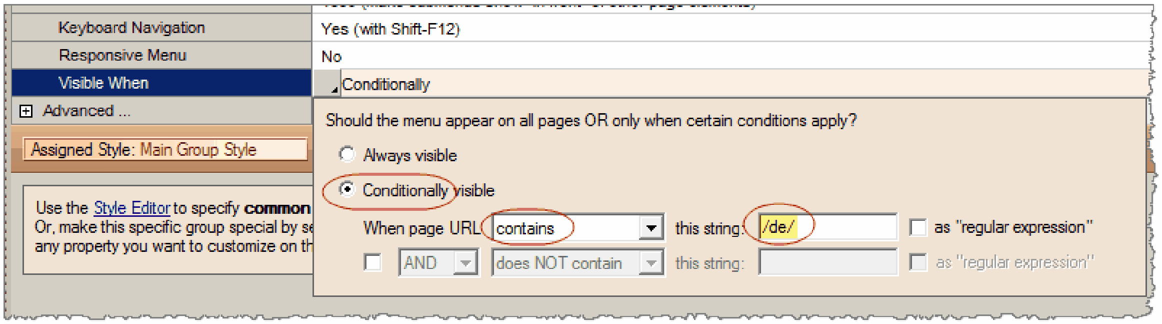
Use the Style Editor to specify **common**
Or, make this specific group special by se
any property you want to customize on th

Should the menu appear on all pages OR only when certain conditions apply?

○ Always visible
● Conditionally visible

When page URL contains ▾ this string: /en/ □ as "regular expression"
□ AND ▾ does NOT contain ▾ this string: □ as "regular expression"

On the **German** menu project (menu-de.js) you set the following condition:



Keyboard Navigation: Yes (with Shift-F12)
Responsive Menu: No
Visible When: Conditionally
⊞ Advanced ...
Assigned Style: Main Group Style

Use the Style Editor to specify **common**
Or, make this specific group special by se
any property you want to customize on th

Should the menu appear on all pages OR only when certain conditions apply?

○ Always visible
● Conditionally visible

When page URL contains ▾ this string: /de/ □ as "regular expression"
□ AND ▾ does NOT contain ▾ this string: □ as "regular expression"

In that way only one of the two linked menus appear on the web page, depending on the page's language implied by the URL.

Note that you can use regular expressions in the matching string. More info on regular expressions.

7.4.16 Property: PopUp

This property is deprecated and replaced by the "Manually Trigger Main Menu or any Group on user action" option of the <Positioning> property.
(You will not see this property in version 5 and above. We retain this help item for reference by users of older versions)

If you are using the old ***awmShowMenu();*** function in your existing projects you do not need to worry, as the new positioning features are backwards compatible with these functions.

To view how the new positioning features are used for popup implementations, click on the online examples below:

http://www.likno.com/examples/demo_popup.html

http://www.likno.com/examples/example_popup_rightclick.html

http://www.likno.com/examples/example_showhidemenu.html

Values:	yes/no
Applies to:	Group (Main Menu only) - Deprecated by the <Positioning> property
Type:	Content Property

See also
How Do I, Replace the default right-click menu used by the browser

7.4.17 Property: Javascript Command on Appear

A javascript code to be executed upon appearance of the Group (Main Menu or submenu).

You can use more than one javascript commands, provided that you separate them with a semicolon (;).

Note! You may use only quotes (") or only apostrophes (') inside the text of the property but **not both**! Using both quotes and apostrophes will cause a javascript error.

Note 2: Javascript Commands are deactivated when previewing your menu inside the Preview Menu window so that they do not produce any javascript errors. Javascript Commands ***do*** work on your web pages when your menu is compiled.

Values:	Text (any javascript command)
Example:	alert("Hello");
Applies to:	Group
Type:	Content Property

7.4.18 Property: Javascript Command on Disappear

A javascript code to be executed upon disappearance of the Group (Main Menu or submenu).

You can use more than one javascript commands, provided that you separate them with a semicolon (;).

Note! You may use only quotes (") or only apostrophes (') inside the text of the property but **not both**! Using both quotes and apostrophes will cause a javascript error.

Note 2: Javascript Commands are deactivated when previewing your menu inside the Preview Menu window so that they do not produce any javascript errors. Javascript Commands

do work on your web pages when your menu is compiled.

Values: Text (any javascript command)

Example: alert("Hello");

Applies to: Group

Type: Content Property

7.4.19 Property: Assigned Style

This is the style that a Group or an Item is assigned from the Style Editor.

The use of styles can help you make global changes to multiple items at once.

For example, you can make similar changes to all your items of your Main Menu Group by simply editing the corresponding style in the Style Editor.

Detailed info on the Assigned Style and all Available Styles

Values: A style from **the six standard styles** of the Style Editor

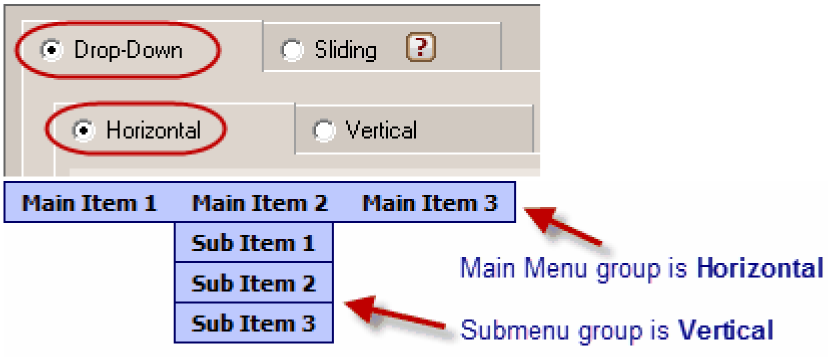
Applies to: Group and Item

Type: Style Property

7.4.20 Property: Type

Defines whether the Group is Drop-Down or Sliding, vertical or horizontal, with how many columns/rows and more options.

Option 1: Drop-Down, Horizontal


● Drop-Down ○ Sliding

● Horizontal ○ Vertical

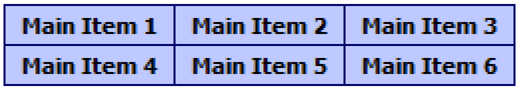
Horizontal groups may also behave as "**Menubars**" (i.e. fitting the entire window width) through the available option (see below).

More Options:

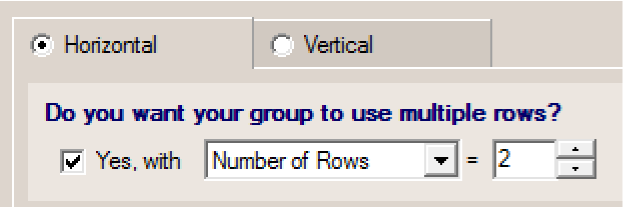
- **"Do you want your group to use multiple rows?"**

  By selecting "Yes, use X rows", your group will use X rows to display the items.

  Example with 2 rows:



Main Item 1	Main Item 2	Main Item 3
Main Item 4	Main Item 5	Main Item 6



● Horizontal ○ Vertical

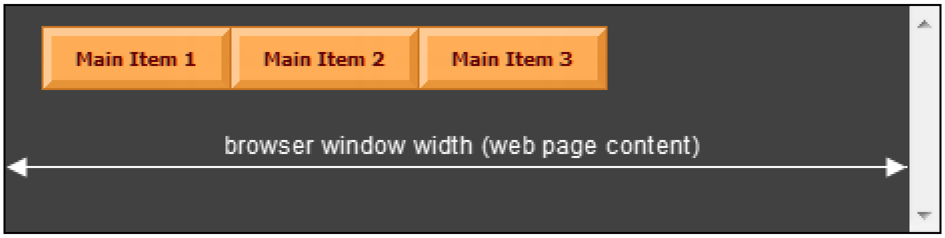
Do you want your group to use multiple rows?

☑ Yes, with Number of Rows = 2

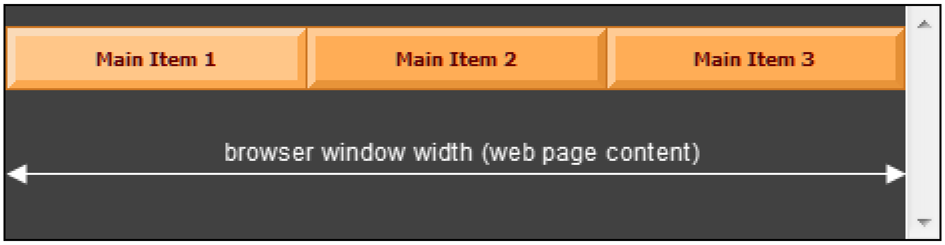
- **"Should the menu function as a Menubar?"**

  By clicking "**Yes**, widen group to fit the entire width of window Browser Window", the group acts like a typical Windows-application menu: horizontal, covering all the width of the window even when the user resizes it.

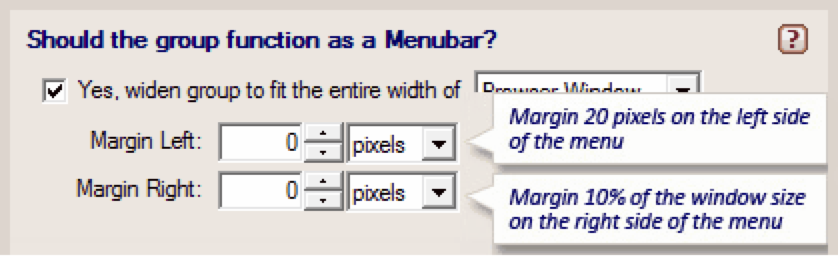
  For example, a non-menubar menu...



...shows like that when its "menubar" option is checked:



You can also add Margin to any of the sides of the menu (Right, Left, or both) counted in pixels or in percentage of the page:



Should the group function as a Menubar?

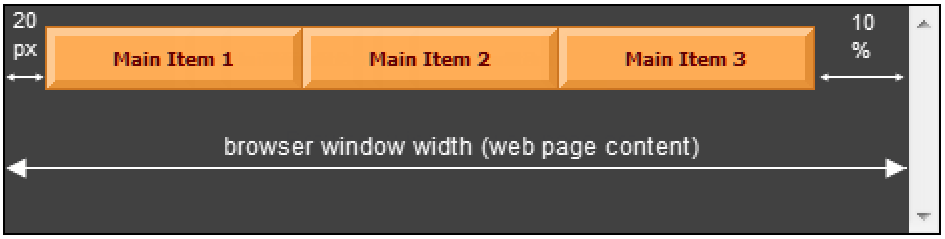
☑ Yes, widen group to fit the entire width of Browser Window

Margin Left: 0 pixels

Margin Right: 0 pixels

Margin 20 pixels on the left side of the menu

Margin 10% of the window size on the right side of the menu

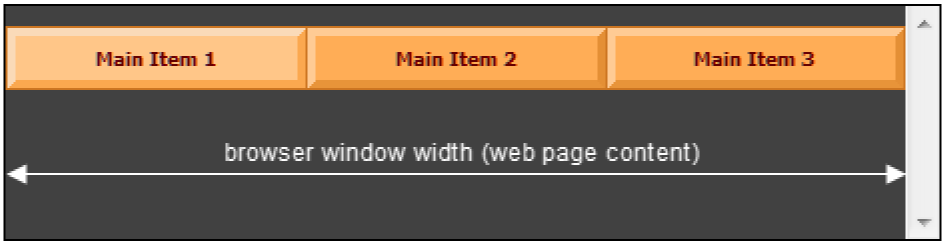


This is a convenient way to center your menu in a centered web site.

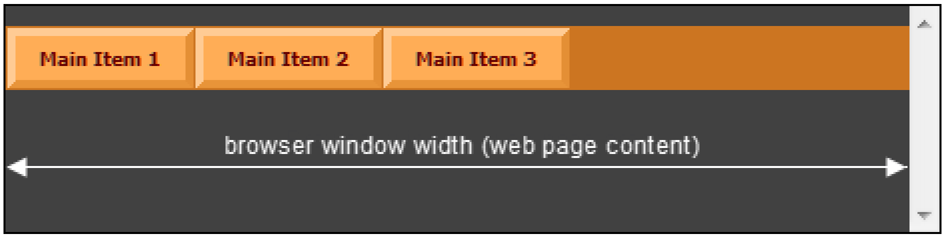
Since version 5.3.916, you can also refer to an element of the page (div, td, etc.) to determine the width of the menu ("menubar"). Just specify its id or class and the menu will fit whatever width that page element has.

- **"Do you want the items to widen to fit their group's width?"**
  (enabled when the "Menubar" functionality above is selected)

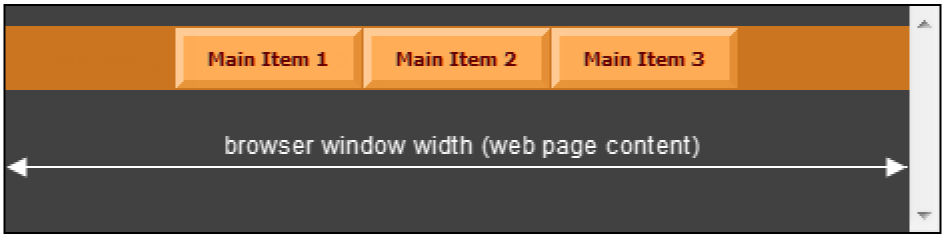
By selecting "Yes, widen items so that they fit the entire group", the items of the menu widen in order to cover the whole size of the menubar.



By selecting "Do not widen items", the items keep their default size. The background color of the menu group covers the rest of the menubar.



You can also choose the alignment of the menu (left, centered, right) inside the menubar.



- **"Should menu affect its positioning element height?"**
  (menu-wide option, actually related to the Main Menu group only)

  "Positioning element" is the **page element** or **image** specified at the <Positioning> property, related to which the menu is positioned on page.

  By selecting "No", the Main Menu will not occupy space inside its positioning element and therefore it will not affect the element's height. The Main Menu will just **float** above it, without pushing (or pulling) the element.

  By selecting "Yes", the Main Menu occupies actual space inside its positioning element, thus forcing the element to get the same height as the Main Menu.

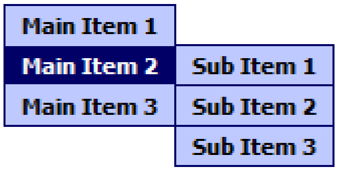
  Notes for the "Yes" setting:

  (A) It has no effect when the Main Menu is positioned "Relative to Window" (it behaves as "No").

  (B) It affects only the **height** of the page element, NOT its width.

  (C) It has no effect if the positioning element is a SPAN (it behaves as "No"). Use a DIV (or a TD, an IMG, etc.) instead.

Option 2: Drop-Down, Vertical

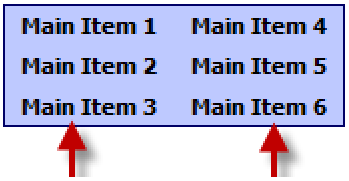


More Options:

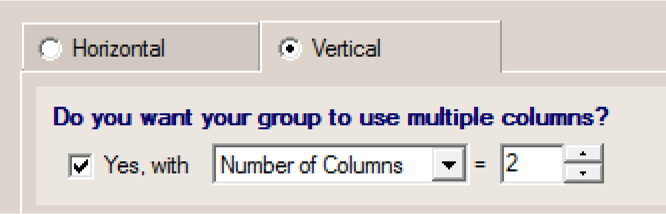
- **"Do you want your group to use multiple columns?"**

By selecting "Yes, use X columns", your group will use X columns to display the items.

Example with 2 columns:



Main Item 1 Main Item 4
Main Item 2 Main Item 5
Main Item 3 Main Item 6



○ Horizontal ● Vertical

Do you want your group to use multiple columns?

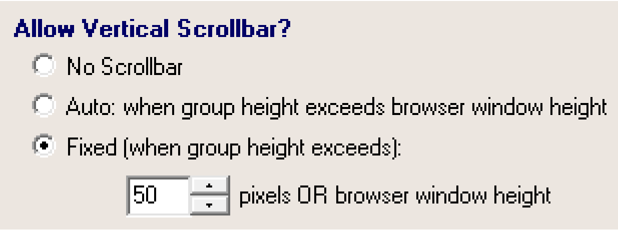
☑ Yes, with Number of Columns = 2

- **"Allow Vertical Scrollbar?"**

By selecting "No Scrollbar" the group will widen to fit all items.

By selecting "Auto: when group height exceeds browser window height", a scrollbar will be added in the menu when its size exceeds the browser window height in order for the viewer to be able to view all the items.
Note that the "Auto" option applies to **Submenu groups** only (it will not apply to the Main Menu Group).

By selecting "Fixed", a scrollbar will be added when the group size exceeds the pixels you have set in the related box.



Allow Vertical Scrollbar?

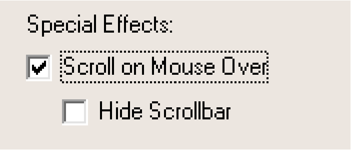
○ No Scrollbar

○ Auto: when group height exceeds browser window height

● Fixed (when group height exceeds):

50 pixels OR browser window height

By selecting to use the scrollbar feature ("Auto" or "Fixed"), additional options are enabled for customization of the scrolling process:



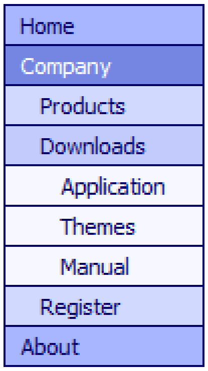
Special Effects:

☑ Scroll on Mouse Over

□ Hide Scrollbar

Option 3: Sliding (vertical only)

(for full info on its settings read the Customize a menu with "sliding" submenu groups topic)



A Sliding group pushes the items of its parent group downwards when opening. In the same manner, it pulls these items upwards when it closes.

This type of menu requires an AllWebMenus license that has the "Sliding Menu Add-in" purchased and activated.

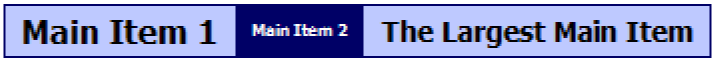
For all information on the settings of this option read the Customize a menu with "sliding" submenu groups topic.

The Uniform Items option

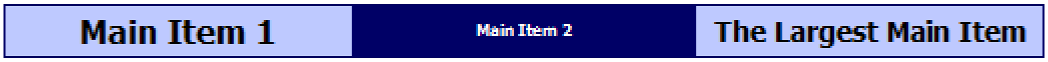
This option allows you to choose whether the size of all items match the size of the biggest item within the same group. The calculation of the largest size includes the size of the Header or Footer too (if they exist).

Example for horizontal menu:

1st menu: The <Uniform Items> property is set to "no"

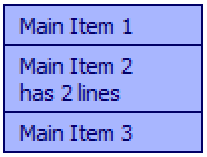


2nd menu: The <Uniform Items> property is set to "yes"
(all items have the same, largest width)

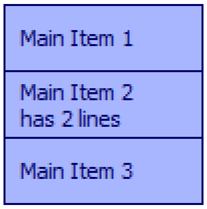


Example for vertical menu:

1st menu: The <Uniform Items> property is set to "no"



2nd menu: The <Uniform Items> property is set to "yes"
(all items have the same, largest height)



Values: Drop-Down, Horizontal (with options)
Drop-Down, Vertical (with options)
Sliding (with options)

Applies to: Group

Type: Content Property

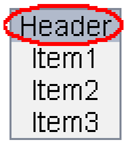
See also:
"Drop-Down vs Sliding" option

7.4.21 Property: Has Header

Specifies whether a group will show a Header or not.

If you select "Yes, Show Header" then a Header will show up at the top of the group.

This is an example of a Header on a menu:



The **content** of the Header is defined by the following content properties:
<Header Text>
<Header Image>
<Header Tooltip>

The **style** of the Header is defined by the style properties that are enabled when you select the "Yes, Show Header" value.

If you do not want to display a Header on the group, you can do so by setting the *<Has Header>* property to "No, Do not show Header" **or** by leaving the <Header Text> and <Header Image> properties blank.

Values: Yes/No

Applies to: Group

Type: Style Property

Lite only: Lite version does not support Header.

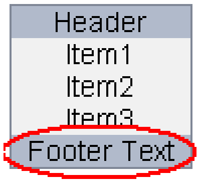
See also
Header Text
Header Image
Header Tooltip
Footer Text

7.4.22 Property: Has Footer

Specifies whether a group will show a Footer or not.

If you select "Yes, Show Footer" then a Footer will show up at the bottom of the group.

This is an example of a Footer on a menu:



The **content** of the Footer is defined by the following content properties:
<Footer Text>
<Footer Image>
<Footer Tooltip>

The **style** of the Footer is defined by the style properties that are enabled when you select the "Yes, Show Footer" value.

If you do not want to display a Footer on the group, you can do so by setting the <*Has Footer*> property to "No, Do not show Footer" **or** by leaving the <Footer Text> and <Footer Image> properties blank.

Values: Yes/No

Applies to: Group

Type: Style Property

See also
Footer Text
Footer Image

[Footer Tooltip](#)
[Header Text](#)

7.4.23 Property: Appears - Where

Specifies **where** the group appears relative to its parent group.

"**Auto**" is the most common value in this property, as it automatically decides where the group appears based on the available space at each side of its parent group.

This is how the "Auto" feature works:

- If parent group is vertical, the group appears at its right side unless it finds no available space and therefore appears at the left side.
- If parent group is horizontal, the group appears below the parent unless it finds no available space and therefore appears above the parent.

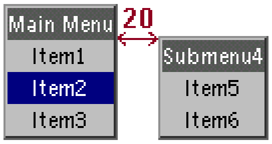
However, you may want to force the group to **always** open on a specific side of the parent group. This can be achieved with the following values:

"Right/Bottom edge of parent group"
Depending on the type of the parent group, the group appears always to the **right** of a **vertical** parent group or **below** a **horizontal** parent group.

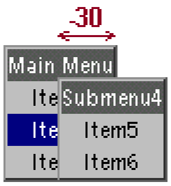
"Left/Upper edge of parent group"
Depending on the type of the parent group, the group appears always to the **left** of a **vertical** parent group or **above** a **horizontal** parent group.

The last parameter specifies the distance of the group relative to its parent group's edge.

For example:



or



Values:	Auto Right/Bottom edge of parent group Left/Upper edge of parent group
Applies to:	Group
Type:	Style Property

7.4.24 Property: Appears - Aligned with

Specifies how the group aligns with its parent item (or group).

"**Parent item**" is the default value in this property as it automatically aligns the submenu group with its parent item, a standard behavior in menus.

However, you are able to use several interesting options with how submenus align and create certain effects:

"Top/Left Edge of parent"
Depending on the type of the parent group, the submenu group aligns with the **top edge** of a **vertical** parent item/group OR with the **left edge** of a **horizontal** parent item/group.

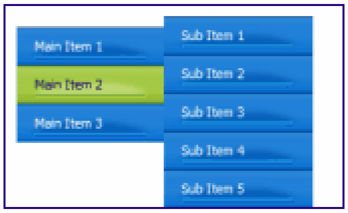
"Bottom/Right Edge of parent"
Depending on the type of the parent group, the submenu group aligns with the **bottom edge** of a **vertical** parent item/group OR with the **right edge** of a **horizontal** parent item/group.

"Centered to parent"
The center of the submenu group is aligned with the center of the parent item/group.

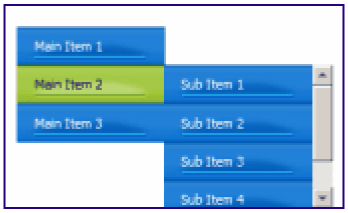
The option **"Alignment Offset"** lets you make minor adjustments to the selected alignment.

The option **"Shifts automatically when group area crosses page boundaries"** determines what happens when the submenu area needs more space than provided by the page.

When "checked" (shifts):



When "unchecked" (does not shift):



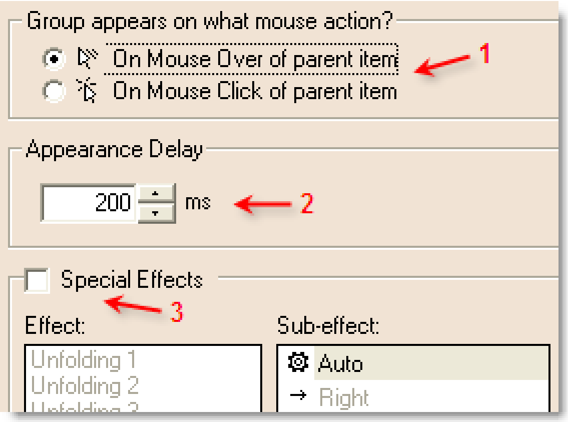
Values: Parent Item
Parent Group
Top/Left Edge of parent
Bottom/Right Edge of parent
Centered to parent

Applies to: Group

Type: Style Property

7.4.25 Property: Appears - How

Specifies **how** the group **appears**.



Group appears on what mouse action?

● On Mouse Over of parent item

○ On Mouse Click of parent item

Appearance Delay

200 ms

□ Special Effects

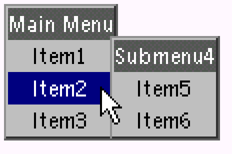
Effect: Unfolding 1, Unfolding 2, Unfolding 3

Sub-effect: Auto, Right

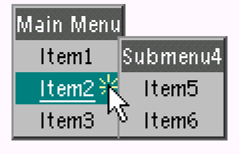
- Arrow [1] shows that the **"Group appears on what mouse action"** functionality.

A group can show up in two ways: when the **mouse is over** its parent item or by **clicking on** its parent item.

Mouse is over the item:



Clicking on the item:



- Arrow [2] shows the **"Appearance Delay"** feature. You can specify how many **milliseconds** should the mouse remain over the parent item in order for the group to appear.

- Arrow [3] shows the **"Special Effects"** feature.

A group can appear using a "special effect".

Unfolding 1, 2 and 3 effects are slightly different versions of the unfolding effect.

Moreover, there are additional effects for the appearance of the submenu in IE5.5+ browsers. These *IE5.5+ only* values make the submenu open using the "Unfolding 2" effect in the non-IE5.5+ browsers.

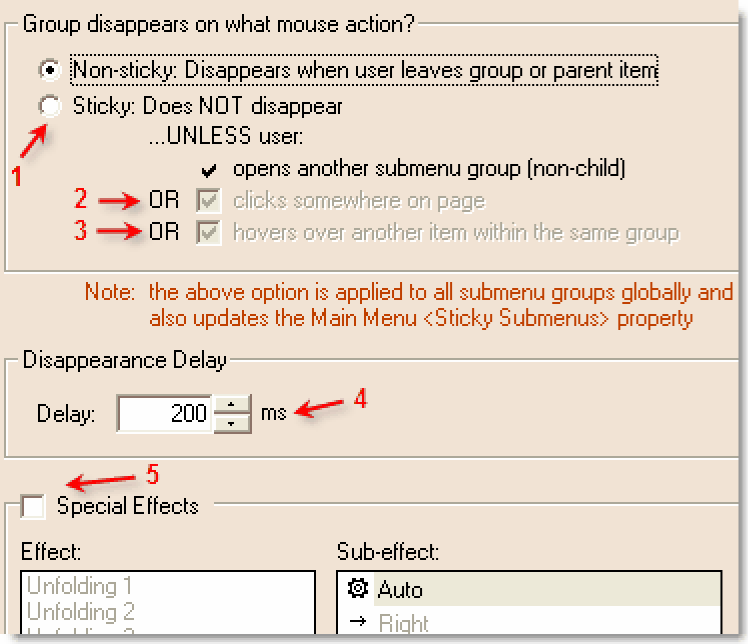
Values: (composite property)

Applies to: Group

Type: Style Property

7.4.26 Property: Disappears - How

Specifies **how** the group **disappears**.



Group disappears on what mouse action?

● Non-sticky: Disappears when user leaves group or parent item

○ Sticky: Does NOT disappear

...UNLESS user:

✓ opens another submenu group (non-child)

OR ☑ clicks somewhere on page

OR ☑ hovers over another item within the same group

Note: the above option is applied to all submenu groups globally and also updates the Main Menu <Sticky Submenus> property

Disappearance Delay

Delay: 200 ms

□ Special Effects

Effect: Unfolding 1, Unfolding 2

Sub-effect: Auto, Right

- Arrow [1] shows that you can specify if all submenus are "sticky" or "non-sticky" within a group property, teamed with other "disappearance-related" values. The Main Menu <Sticky Submenus> property is still used for backwards compatibility and has exactly the same effect. Note that this particular option **applies globally to all submenus** of the project, not only to the selected submenu.

- Arrows [2] and [3] show some new interesting variations of "sticky" submenus.

> By unchecking the default check on arrow [2] you can have the last opened submenu always visible, regardless of what the user does on the rest of the page.
>
> By also unchecking the default check on arrow [3] you can simulate a TAB-like behavior for the menu, as at least one submenu will **always** remain open no matter what happens on page or menu (see "XP Tabs" online example).

- Arrow [4] shows the "Disappearance Delay" feature. You can specify how many milliseconds should the mouse move away from the group in order for the group to disappear. This option gives extra time to the group to remain open if the mouse accidentally slips out of the group and gets back again within the time limit specified.

- Arrow [5] shows the "Special Effects" feature. You can specify an effect for the disappearance of the group. Combined with the "Appearance Method" of the <Appears - How> property, you can have interesting effects when navigating from submenu to submenu. A group can disappear using a "special effect". Unfolding 1, 2 and 3 effects are slightly different versions of the unfolding effect. Moreover, there are additional effects for the disappearance of the submenu in IE5.5+ browsers. These *IE5.5+ only* values make the submenu disappear using the "Unfolding 2" effect in the non-IE5.5+ browsers.

Values:	(composite property)
Applies to:	Group
Type:	Style Property

7.4.27 Property: Opacity

Makes the group appear semi-transparent.

A value of 100 keeps the group fully opaque (default).
A value of zero makes the group fully transparent.

<Group Opacity> combined with the <Item Floating Image> property (if used) note:
If part of the floating image shows outside the item’s Group, make sure that the **Group <Opacity> is 100%** otherwise Internet Explorer does not display this part of the image. This is not important if you do not use any floating images in your project

Values:	From 0 to 100
Applies to:	Group
Type:	Style Property

7.4.28 Property: Group Background Color

Fills the background of the group with a color.

If "none" is selected and no background image is selected (i.e. <*Background Tile Image*> is set to "No Image"), the background looks transparent. See more on background layers.

If a *<Background Tile Image>* is selected, the entire background gets covered by the tiled image instead.

You can actually see the selected color along with its respective hexadecimal code. You can either enter the hexadecimal code directly or pick a color from the color dialog box which appears when clicking on the color box.

Values:	#000000 (Black) - #FFFFFF (White)
Applies to:	Group
Type:	Style Property

See also
The different background layers

7.4.29 Property: Group Background Tile Image

Displays an image on the background of the group.

This image is always **tiled** on the background.

If "No Image" is selected, the entire background get's covered by the group's <Background Color> instead. See more on background layers.

Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values: An image file of the .gif, .jpg or .png type.

Applies to: Group

Type: Style Property

See also
The different background layers

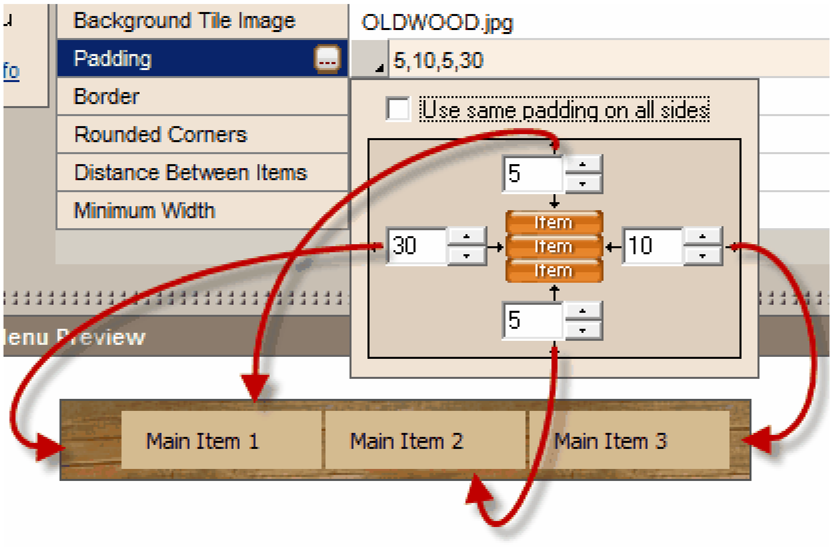
7.4.30 Property: Group Padding

A number of pixels added between the items and the group’s edges (borders).

You can adjust the padding of each side all together or **even separately!**
(see image below for different padding per side)

Values: positive numbers in pixels

Example: 5,10,5,30 (top, right, bottom, left)



Background Tile Image: OLDWOOD.jpg

Padding: 5,10,5,30

□ Use same padding on all sides

5

30

10

5

Applies to: Group

Type: Style Property

7.4.31 Property: Group Border

Specifies the style of the group's border.

This is the border that covers the whole group area, including the group's Header or Footer (if exist). It is **not** a border that applies to each item contained in the group. Instead, items have their own <Item Border> property which can co-exist with the group border.

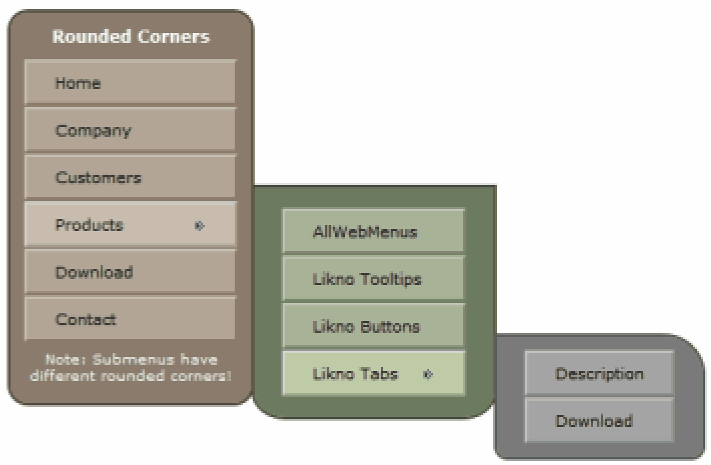
Values: [none, solid, inset, outset], [color], [width]

Applies to: Group

Type: Style Property

7.4.32 Property: Group Rounded Corners

Specifies if the corners will look rounded and to which degree (through the X and Y radius values).



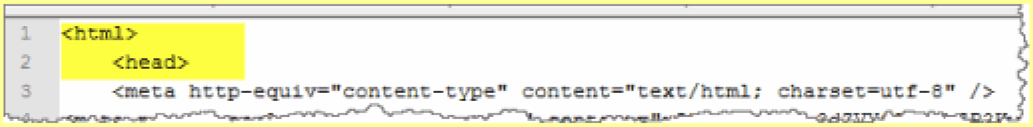
You can adjust the roundness of each corner all together or **even separately**. (see submenus in above menu)

Do not forget that in most cases you will also need to **add padding** to your group, either through the separate Group Padding property or through the attached padding values that are a shortcut to that property.

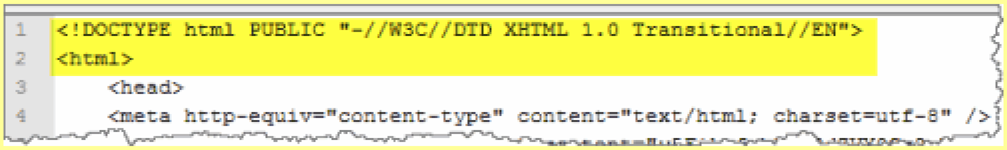
Important! This capability is based on the **border-radius** property of the **CSS3** specification, which is available in **I** **3.5+, Safari (Webkit 522+), Chrome and others**. All earlier browsers (such as IE8 or below) **do not support CSS** problem as the script degrades this feature gracefully on those browsers by **ignoring** its settings and showing all c

Internet Explorer 9 also requires that your pages use a full DOCTYPE declaration, otherwise it cannot support the

For example, the following pages will **not** show rounded corners in IE9...

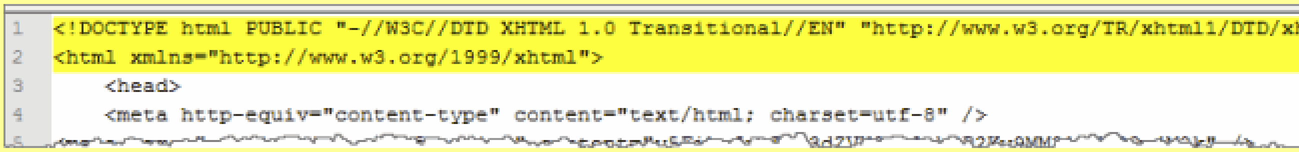


```
<html>
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

...but the following pages **will** show rounded corners in IE9:



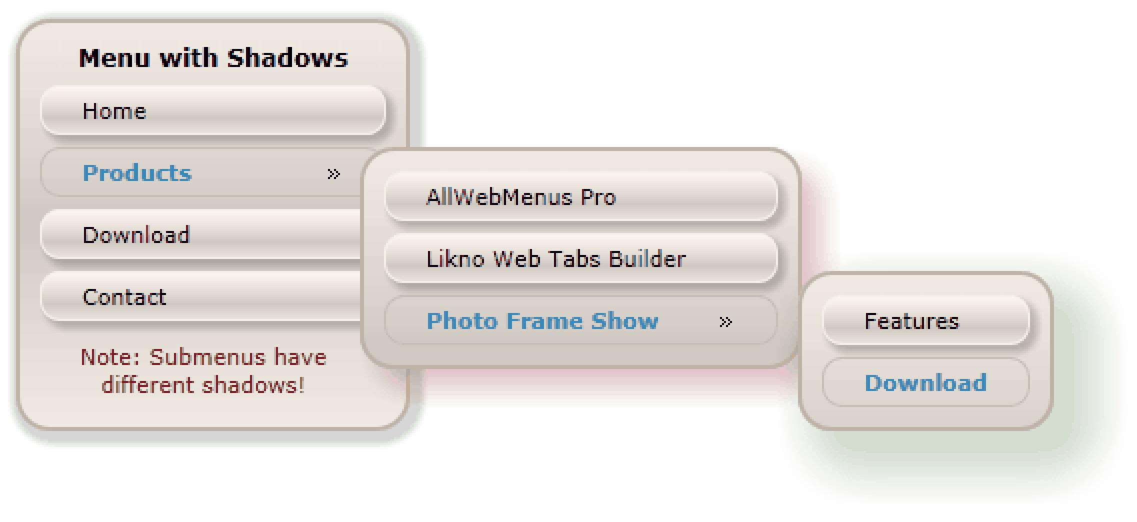
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/x
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

You can find more info on the **border-radius** capabilities here:
http://www.css3.info/preview/rounded-border/

Values:	Color: #000000 (Black) - #FFFFFF (White) Offset X, Offset Y, Blur, Spread: number in pixels
Applies to:	Group
Type:	Style Property

7.4.33 Property: Group Shadow

Specifies if the group has a shadow and how this shadow looks like.

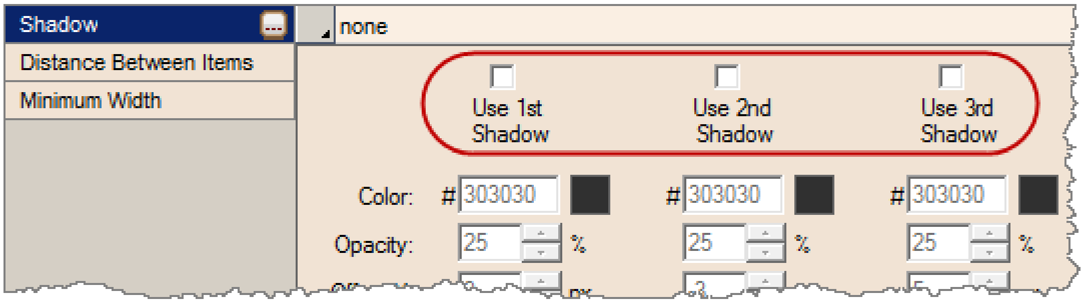


You can specify several appearance options for a shadow:

color, opacity, offsets, blur, spread

Through the styles (or style customizations) you can also use different shadows on groups (see submenus in above menu).

You can mix up to **3 different shadows into a single shadow** to create unique effects:



Shadow	none
Distance Between Items	
Minimum Width	

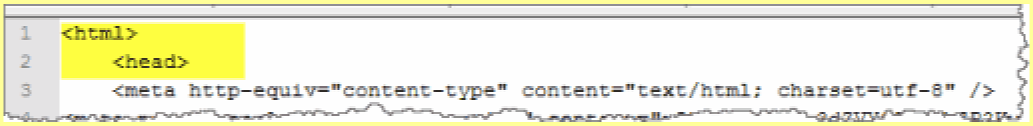
□ Use 1st Shadow □ Use 2nd Shadow □ Use 3rd Shadow

Color:	# 303030	# 303030	# 303030
Opacity:	25 %	25 %	25 %

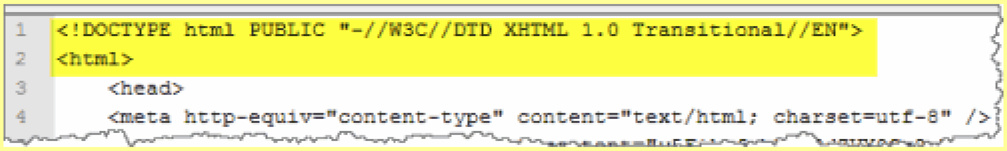
Important! The "shadow" feature is based on the **box-shadow** property of the **CSS3** specification, which is availab **Firefox 3.5+, Safari (Webkit 522+), Chrome and others**. All earlier browsers (such as IE8 or below) **do not supp** not a problem as the script degrades this feature gracefully on those browsers by **ignoring** its settings and showing shadow.

Internet Explorer 9 also requires that your pages use a **full DOCTYPE declaration**, otherwise it cannot support th property (see example below).

For example, the following pages will **not** show shadows in IE9...

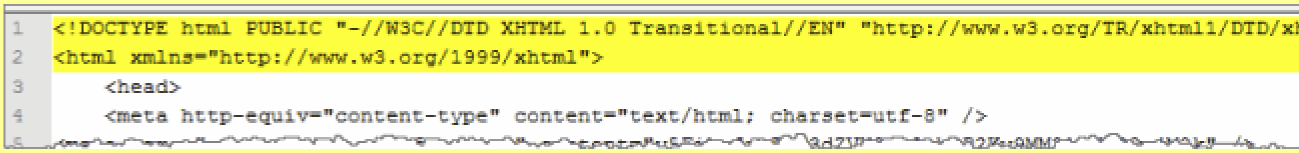


```
<html>
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

...but the following pages **will** show shadows in IE9:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/x
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

You can find more info on the **box-shadow** property capabilities here:
http://www.css3.info/preview/box-shadow/

Values:	Color, Opacity, Offset X, Offset Y, Blur, Spread
Applies to:	Group
Type:	Style Property

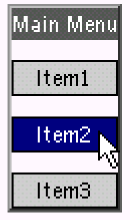
See also
Item Shadow

7.4.34 Property: Distance Between Items

Specifies the distance between the adjacent items of a group.

Most conventional menus will have this set to zero.

By entering a non-zero value you create gaps between the items. In that way, you can have a "ladder" effect as shown below:



In this example, the background of items is grey while the background of the group is transparent (not white).

Values: A positive number in pixels

Applies to: Group

Type: Style Property

7.4.35 Property: Group Minimum Width

This is the **preferred** width of the group, implying that the group should have a size of **at least** as many pixels wide as those specified.

The **actual** group's width may be larger than the preferred width, as this will depend on the rendered size of the group's content (text or images). If the contents are too large they will "push" the group's width to a larger value than the preferred one, similar to what happens to the size of any <td> if its content is larger than its specified width.

In a vertical menu, new lines (<br>) are entered to any items containing "multiple-words" content, when this content pushes the group's width to be larger than the specified one.

Values: A positive number in pixels, enter zero for not specifying an exact width

Default value: 0 (zero) - Use zero if you want the width to be automatically determined when the menu is rendered.

Applies to: Group

Type: Style Property

7.4.36 Property: Item Text

Specifies the text or HTML shown on the Item. It can also be combined with the <Image> property.

Note: If you want to show an image instead of text on the item, you can do so by choosing the "Image only" value at the "Image Placement" option of the <Image> property. In that case and although blank text is not rejected, it is useful to name all items so that they can be easily identified in the Menu Structure Pane on the left side.

There are three modes you can use to show content on the item.

1. Text Only Mode

The item shows the text exactly as you enter it (except for “custom variables” which are substituted by their values).
HTML tags, multiple spaces, line feeds, etc. are shown as is.

Important note for special extended characters:
It appears that some special characters (eg: Chinese) cannot be inserted or displayed properly inside the text fields. In this case, please do the following:

1. Type the words you want to be shown in the <Text> property into a text editor (which supports the special extended characters).

2. Paste them into the <Text> property. The words may look strange there but do not worry!

3. When compiling the menu and uploading it on the server, the special characters will be displayed correctly.

2. HTML - Formatting Mode

The item treats the text as HTML, for appearance purposes only.
(except for “custom variables” which are substituted by their values)

HTML tags are rendered, multiple spaces and line feeds are ignored (use &nbsp; or <br> instead).

Mouse handling on HTML tags is ignored and passed over to related AllWebMenus properties of the [Mouse Over] and [Mouse Click] states.

For example:

<a href=”http://www.likno.com”>Back to <b><i>Home</i></b></a>

is formatted as Back to ***Home***

but the link is not clickable (the <Link> AllWebMenus property is used instead).

Note: **Relative paths** in attributes (such as “href”, “src”, etc.) should be relative to the location of the actual pages where the menu appears. Relative paths will not render correctly on Preview Pane but will render correctly on actual pages. Still, there is a way for the Preview Pane to also show them: you can add these assets into a "*projectname*-assets" folder, located in the same folder where the project is located (eg: *"menu-assets"*).

3. HTML - Full Mode Online Example

The item treats the text as HTML, both for appearance and functional purposes.
(except for “custom variables” which are substituted by their values)

Contrary to the “HTML - Formatting Mode”, this mode gives you full control through the HTML code (i.e. mouse handling is yours, all Content Properties except <Has Separator> are ignored).

You can use this option to create anything, from a simple working link, up to forms, flash objects, search boxes, even full-scale web pages (recommended only for professionals – no support provided).

View detailed online example at http://www.likno.com/examples.html?example=htmlmenu

Note that the “HTML - Full Mode” option can be applied only through the **[Normal] state** of the <Text> property.
This option disables all Content Properties (except <Has Separator>), so that you have “full” control on your item’s content through your own HTML code. This option also disables the <Foreground Image> and <”Has Submenu” Image> Style Properties.

You may also choose to ignore (disable) all Style Properties for the selected item if you want to use your own style formatting through your own HTML code.
For example, you may want to specify border or padding attributes through your own code and not through the interface.
In most cases you do not need to use this, as your own HTML code formatting overrides the Style Properties values specified through the interface.

Note: **Relative paths** in attributes (such as “href”, “src”, etc.) should be relative to the location of the actual pages where the menu appears. Relative paths will not render correctly on Preview Pane but will render correctly on actual pages. Still, there is a way for the Preview Pane to also show them: you can add these assets into a "*projectname*-assets" folder, located in the same folder where the project is located (eg: *"menu-assets"*).

Values: Text, HTML
Unicode text fully supported (all languages and symbols)

Example1: Company Info
Example2: <b>More</b> Info
Example3: More <br> Info (very useful to break a long item name into more lines, when "HTML - Formatting Mode" is selected)

Applies to: Item

Type: Content Property

See also
Item Image
Custom Variables

7.4.37 Property: Item Image

Displays an image on the Item.

- You can use the **Image Placement** option to specify the relative positioning between the image and the item's text or choose to display the image only.

- You can use the **Image Margin** option to specify margins on the image, allowing you to choose any white space you like at its sides.

In case you want to have an <alt> attribute for tooltip text related to the image, there is the <Tooltip> property for this purpose which refers to the entire item.

Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values:	An image file of the .gif, .jpg or .png type
Applies to:	Item
Type:	Content Property

See also
Item Text
Item Foreground Image

7.4.38 Property: Item Link

Specifies the web page to be opened when clicking on the item.

You can use both **absolute** and **relative** paths.

Absolute paths

The application considers the Link as an absolute path **only if the ://** substring is found in the string, such as:

http://
https://
ftp://
mms://
mailto:
file:////servername/ (if you access a server)
file:///driveletter|/ (if you access a drive)
or **file:///**driveletter:/ (if the above drive syntax is not supported by your network you may need to try both and keep the one that works)

Valid examples of absolute paths:
http://www.likno.com
http://www.likno.com/faq.html
ftp://www.likno.com/downloads/AllWebMenus.exe
mailto:support@likno.com
file:////server1/testarea/test.html (for intranet using a server)
file:///C|/testarea/test.html (for intranet using a hard drive) **or**
file:///C:/testarea/test.html (if the above syntax is not supported by your network)

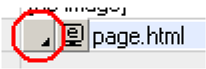
Invalid examples of absolute paths:
www.likno.com
likno.com
(these strings will be parsed as relative paths and the correct URL will **not** open)

Relative paths

Both **menu-relative** and **root-relative** paths are allowed here.

The former is related to the folder where the **compiled menu** is located while the latter is related to the folder where the **site root** is located (and always starts with a slash "/"). While all types of paths successfully open their assigned pages when viewed on the server, the root-relative paths cannot resolve their assigned pages when viewed locally because Windows considers the C: drive as being the root. On the contrary, menu-relative paths do open their pages correctly when viewed locally. Therefore, as there are no important differences between the two, **it is advisable to use menu-relative paths**.

The best way to choose your linked page from within your local web site structure is to press the related button shown below:



page.html

As an example, let's say that the local root folder of your site is the **c:\mysite** folder and that the contents of your web site are:

```
| ---------------------------------------------- |
| c:\mysite\index.html                |
| c:\mysite\company.html              |
|          ...                        |
| c:\mysite\products\product.html  |
| ---------------------------------------------- |
```

<u>Example 1</u>:
Let's say that you compile your menu as **c:\mysite\mymenu.js**

For the item that opens the product page:
- The menu-relative URL is: **products/product.html**
- The root-relative URL is: **/products/product.html**

For the item that opens the company page:
- The menu-relative URL is: **company.html**
- The root-relative URL is: **/company.html**

<u>Example 2</u>:
Let's say that you compile your menu as **c:\mysite\menufolder\mymenu.js**

For the item that opens the product page:
- The menu-relative URL is: **../products/product.html**
- The root-relative URL is: **/products/product.html**

For the item that opens the company page:
- The menu-relative URL is: **../company.html**
- The root-relative URL is: **/company.html**

Anchors

A menu item can also link to a paragraph within a page:
Enter **http://www.mysite.com/info.html#anchor** in the <Link> item property and the menu can be used to navigate inside the **www.mysite.com/info.html** web page.

You may also insert just an anchor (eg. **#top**) in the <Link> property, without a specific URL. This is useful when you have many different pages using the same anchor and you want a menu item to serve the purpose of navigating to this anchor within the loaded page.

As implied by its definition, the property is only enabled on the Mouse Click state.

Notes: **-** Always use **slashes** instead of backslashes
(example: ../users/~user10/user10.html)
- Don't forget that the URL is case-sensitive on unix servers!

Values: Text

Example1: index.html
Example2: index.html#anchor
Example3: http://www.likno.com
Example4: mailto:support@likno.com

Applies to: Item

Type: Content Property

See also
Custom Variables

7.4.39 Property: Item Link Target

Specifies the frame or window where the page specified by the <Link> property opens.

If you do not use frames then just **leave this field blank** OR choose **_blank** to open the <Link> into a new browser window.
If you use frames, you can also find additional information on how to properly setup a cross-frame menu at the following topic:
Commands -> Tools -> Project Properties -> Compile Properties: Cross-Frame Support

Values:

Leave it blank (none) to open the <Link> page into the same frame or window where the menu is located (**default** for non-framed pages).

Choose **_blank** to open the <Link> page into a new browser window.

Choose **_top** to open the <Link> page in the whole area of the current browser window (not to a specific frame).

Choose the **name of the frame or iframe** where you want the <Link> page to open in (this is **not** the filename!).

OR use the <Open Modal Window> property to show any type of content in a **modal window** that appears above the loaded page.
(see online examples on how usable this can be for your websites)

Example with Frames:

Let's assume that we have the following Frameset Definition inside the source code of the home page:

```
<frameset framespacing="0" border="0" cols="171,*" frameborder="0">

  <frame name="left" target="main" src="left_page.htm" scrolling="auto">
  <frame name="main" src="main_page.htm" target="_self">

  <noframes>
  <body>
  <p>You need a browser that supports frames to see this site </p>
  </body>
  </noframes>

</frameset>
```

Let's also assume that we have placed our menu at the left_page.htm page (opening at the frame named "**left**"). We want the menu items to open their pages specified by their <Link> property inside the frame named **"main"**.

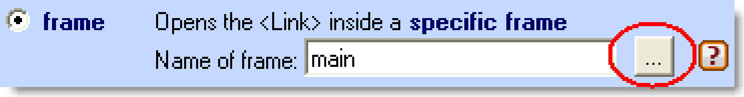
As we can see from the <frameset> tag, the frame where the main_page.htm opens has been assigned the following name:

main

This is exactly what you need to enter into the **<Link Target> property** of the menu items. Do not get confused and use the content of the **src** attribute (eg: main_page.htm); this will **not** work. You need to use the content of the **name** attribute (eg: **"main"**).

The same logic applies for the **iframe** option.

A wizard can also help you choose the correct value by pressing the related button shown below:



● **frame** Opens the <Link> inside a **specific frame**

Name of frame: main

(you are not required to use the wizard, you can simply type the frame name in the text field if you already know the name)

Note! If you use the **frame or iframe** option, the Menu Preview window will ignore it so that no javascript errors are produced. The <Link Target> property **does** work when your menu is compiled and linked to your **actual** pages.

Values: Text

Applies to: Item

Type: Content Property

See also

The <Link> property

Commands -> Tools -> Project Properties -> Compile Properties: Cross-Frame Support

7.4.40 Property: Item Status Bar Text

Specifies the text appearing on the browser's status bar when the mouse passes over the Item.

Use this property to display a small status bar text description for each item.

You can use the **%Link%** parameter (default) to show the contents of the <Link> property on the status bar.
This parameter can be used combined with the rest of the text. For example, if the <Link> property has the value "orderpage.html" then you can use the following value on <Status Bar Text>:

http://www.mysite.com/%Link%

This will result in the following status bar text when the mouse is over the item:

http://www.mysite.com/orderpage.html

As implied by its definition, the property is only enabled on the Mouse Over item state.

Important Note:

- Status bar text manipulation is not working for menus viewed in Internet Explorer under the default security settings (internet zone), while websites added in “local intranet” and “trusted sites” should work properly. This restriction was introduced in Internet Explorer 7 as part of the security work to reduce spoofing and ensure that content on the internet cannot directly influence the area of the browser outside the HTML rendering area.

- The same restriction applies also to Firefox 2 and above. The user needs to have the following option enabled: "Tools -> Options -> Content -> Advanced -> Change status bar text" (this may be in a different location depending on the firefox version used).

Values:	Text Unicode text fully supported (all languages and symbols)
Applies to:	Item
Type:	Content Property

See also
Custom Variables

7.4.41 Property: Item Tooltip

Text that appears above the item when the user pauses the mouse pointer over the item.

The style of the tooltip can be changed through the Item Tooltip Style property. Through that property you can use a stylish tooltip produced with the **separate** Likno Web Tooltips Builder product.

Values:	Text Unicode text fully supported (all languages and symbols)
Applies to:	Item (on [Mouse Over] state only)

Type: Content Property

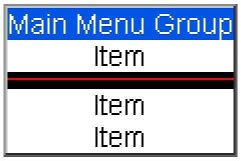
See also
Custom Variables

7.4.42 Property: Item Has Separator

Specifies whether a separator bar exists between the item and its following items.

Note that the separator uses the background color of the group; not the background color of the item.

For example, the menu below uses a red separator on its 1st item. While the items have a white background the group has a black background. This is why the background of the separator is black.



Values: [width], [color], [shadow color]

Applies to: Item

Type: Content Property

7.4.43 Property: Item Javascript Command

Specifies a Javascript Command to be executed when the item comes into a specific state.

Normal state:
It functions like a [Mouse Out] state. The Javascript Command will be executed when the mouse **moves away from** the item's area.

Mouse Over state:
The Javascript Command will be executed when the mouse **passes over** the item's area.

Mouse Click state:
The Javascript Command will be executed when the mouse **clicks on** the item's area. This does not cancel the <*Link*> property. You can set both a <Link> to open **and** a Javascript command to be executed when clicking on the Item.

You can use more than one javascript commands, provided that you separate them with a semicolon (;).

Note! You may use only quotes (") or only apostrophes (') inside the text of the property but **not both**! Using both quotes and apostrophes will cause a javascript error.

Note 2: Javascript Commands are deactivated when previewing your menu inside the Preview Menu window so that they do not produce any javascript errors. Javascript Commands ***do*** work on your web pages when your menu is compiled.

Values: Text (any javascript command)

Example: alert("Hello");

Applies to: Item

Type: Content Property

7.4.44 Property: Item Open Modal Window

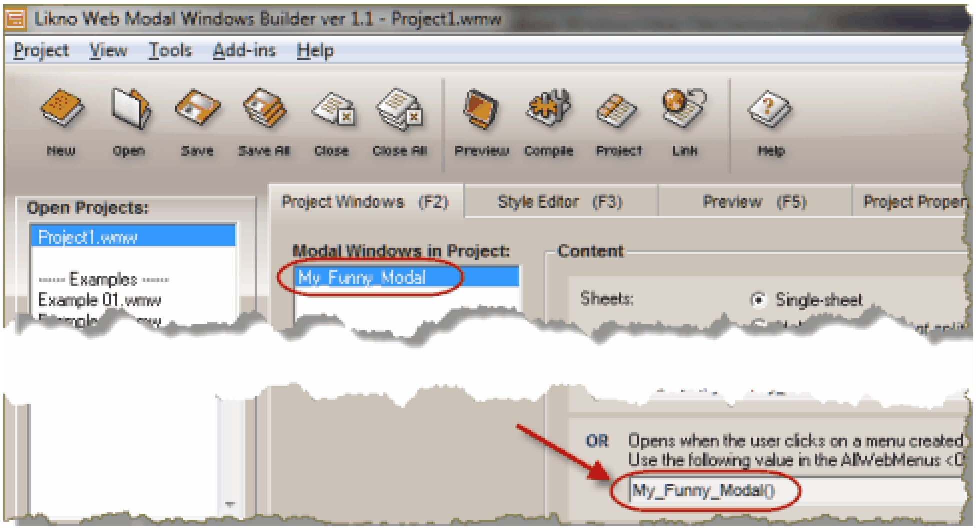
Specifies a "Modal Window" (popup window/dialog) to open when clicking on the item.

The "Open Modal Window" property offers an easy integration between your menu and modal windows produced with the **separate** Likno Web Modal Windows Builder product (in order to create/compile the modal window you need to have a license of that software too).

A modal window is used to display additional content on a new page layer (window) on top of the loaded content. Its main characteristic is that it forces the user to interact only with it, therefore preventing other page actions unless the user closes the window. see examples

How?

1. Open and use **Likno Web Modal Windows Builder** to create, compile and link a modal window project to your pages. Copy the value shown here for the selected modal window:



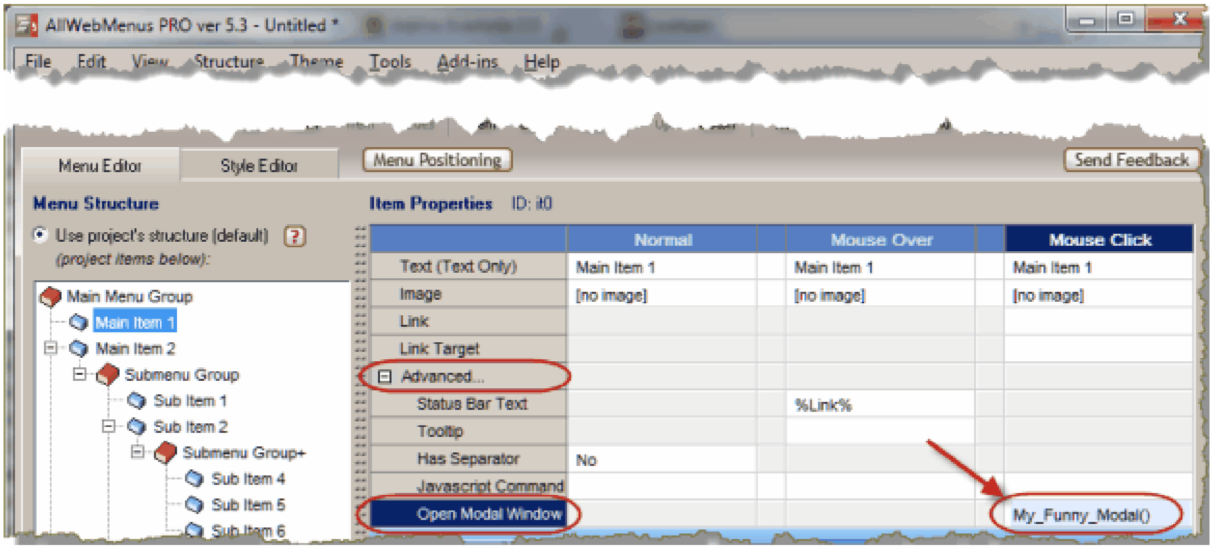
Sheets: ● Single-sheet

OR Opens when the user clicks on a menu created
Use the following value in the AllWebMenus <D

My_Funny_Modal()

2. Load your menu project in **AllWebMenus** (build #792 or above).

3. Choose the <**Open Modal Window**> property under the "Advanced" section of your preferred **Item** property. Paste the copied value there:

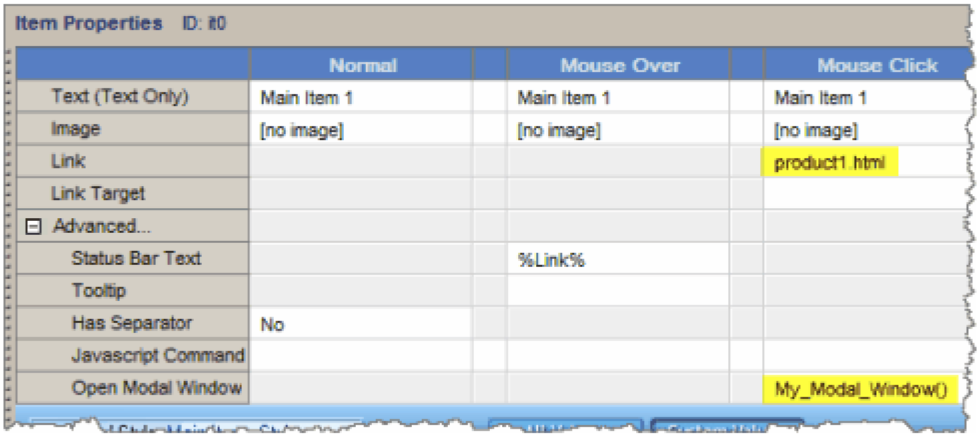


	Normal	Mouse Over	Mouse Click
Text (Text Only)	Main Item 1	Main Item 1	Main Item 1
Image	[no image]	[no image]	[no image]
Link			
Link Target			
⊟ Advanced...			
Status Bar Text		%Link%	
Tooltip			
Has Separator	No		
Javascript Command			
Open Modal Window			My_Funny_Modal()

4. This is all!

Just make sure that the **compiled modal project** (the .js file of a .wmw project) is also linked to the web page where your menu appears ("clicking" on the menu item will do nothing in pages where the modal project is not linked).

Combined capability!
You can **use both the <Link> and "Open Modal Windows" properties together!**



	Normal	Mouse Over	Mouse Click
Text (Text Only)	Main Item 1	Main Item 1	Main Item 1
Image	[no image]	[no image]	[no image]
Link			product1.html
Link Target			
⊟ Advanced...			
Status Bar Text		%Link%	
Tooltip			
Has Separator	No		
Javascript Command			
Open Modal Window			My_Modal_Window()

This means that the modal window can show the page specified in the <Link> property (eg: *product1.html*), if this is not empty.

This is how it works:

What is the value of the <Link> property?

Blank -> Result: The modal window shows its own "content-related" value (the one specified within its project).

Page (*product1.html*) -> Result: The modal window shows the linked page instead (disregards its own "content-related" value).

Notes:

- The modal window cannot be shown within the AllWebMenus "Menu Preview" pane (as this requires external files), but it will be shown on your actual pages (provided that the modal window project exists there).

- If you use more than one modal window in your page, make sure that you give different names to these windows, otherwise you will have a conflict.

Values:	Text *(specified by the Likno Web Modal Windows Builder project)*
Applies to:	Item
Type:	Content Property

7.4.45 Property: Item Content Alignment

Specifies how the item's content aligns with the Item area as a whole.

The **content** consists of both the item's text (<*Item Text*>) and the item's image (<*Item Image*>).

This property should not be confused with the **Image Placement** option of the <Item Image> property which specifies the relative positioning between the item's image and its text.

Values:	Left, Center, Right
Applies to:	Item
Type:	Style Property
Lite only:	Images are not supported by the Lite version

See also
The two kinds of Alignment

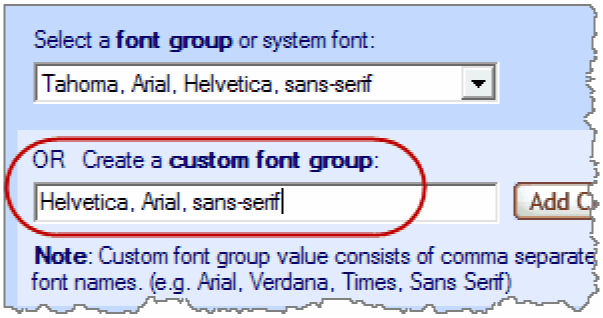
7.4.46 Property: Item Text Font

Specifies the font of the item's text.

The selection box displays several generic font groups (font combinations) along with any font installed on your system.

For cross-platform compatibility, it is **advisable to use the generic font groups** instead of any system-specific font. For example, if you just use "Arial" and a specific platform does not have the Arial font, then the menu will have to use the browser's **default** font.

You may also create custom font groups (combinations), by typing the names of the fonts separated by commas (with the first font having priority over the second one, etc.):



Select a **font group** or system font:

Tahoma, Arial, Helvetica, sans-serif

OR Create a **custom font group**:

Helvetica, Arial, sans-serif

Note: Custom font group value consists of comma separate font names. (e.g. Arial, Verdana, Times, Sans Serif)

Values: Generic combinations and all fonts installed on your system

Applies to: Item

Type: Style Property

7.4.47 Property: Item Text Size

Sets the font size of the item's text.

Allows you to use either CSS sizing, browser-relative sizing or percentage sizing.
You need to click the ◢ button to set values for this property.

We strongly recommend that you use CSS sizing, as this is not affected by the user's preferences.

Values: CSS or browser-relative values or percentage

Applies to: Item

Type: Style Property

7.4.48 Property: Item Text Decoration

Specifies if the item's text is displayed **bold**, **italic, underlined, overline, strikethrough** or **small-caps.**

Values: bold, italic, underlined, overline, strikethrough, small-caps

Applies to: Item

Type: Style Property

7.4.49 Property: Item Text Color

Specifies the color of the item's text.

You can actually see the selected color along with its respective hexadecimal code. You can either enter the hexadecimal code directly or pick a color from the color dialog box which appears when clicking on the color box.

Values:	#000000 (Black) - #FFFFFF (White)
Applies to:	Item
Type:	Style Property

7.4.50 Property: Item Padding

A number of pixels added between the item's content and its edges.

Note that **content** includes both the item's text and image (if any).

Values:	positive numbers in pixels
Example:	0,10,0,10 (top, right, bottom, left)
Applies to:	Item
Type:	Style Property

7.4.51 Property: Item Text Margin

A number of spaces (&nbsp) added just before and after the <Item Text> property.

This property is useful when you want to add "padding" to your item's text at the horizontal axis only. Remember that the <Item Padding> property can also add pixel-based padding at both the horizontal and vertical axis.

Values:	a positive number of spaces (&nbsp)
Example:	2,2
Applies to:	Item
Type:	Style Property

7.4.52 Property: Item Border

Specifies the style of the item's border.

This property refers to the item only and should not be confused with the <*Group Border*> which applies to the whole group where the item belongs.

Values: [none, solid, inset, outset], [color], [width]

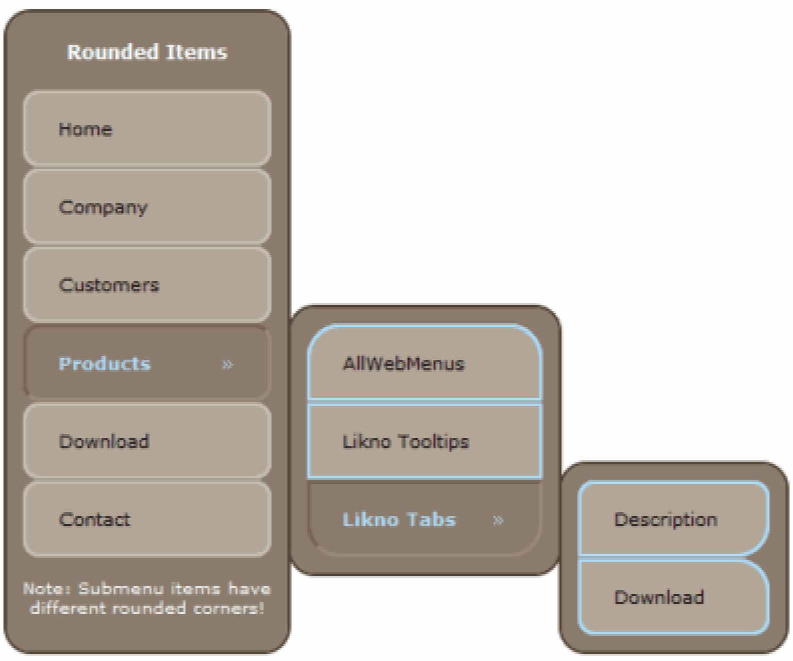
Applies to: Item

Type: Style Property

See also
Group Border

7.4.53 Property: Item Rounded Corners

Specifies if the corners of the item will look rounded and to which degree (through the X and Y radius values).



You can adjust the roundness of each corner all together or **even separately** (see submenu items in above menu).

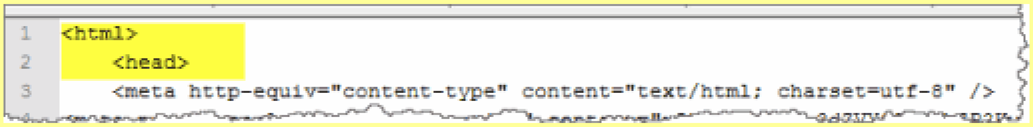
Do not forget that in most cases you will also need to **add padding** to your item, either through the separate Item Padding property or through the attached padding values that are a shortcut to that property.

Important! This capability is based on the **border-radius** property of the **CSS3** specification, which is available in **I** **3.5+, Safari (Webkit 522+), Chrome and others**. All earlier browsers (such as IE8 or below) **do not support CSS**

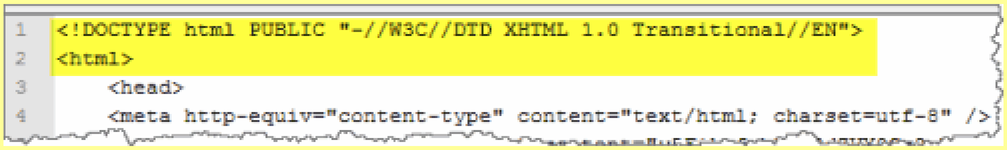
problem as the script degrades this feature gracefully on those browsers by **ignoring** its settings and showing all c

Internet Explorer 9 also requires that your pages use a full DOCTYPE declaration, otherwise it cannot support the

For example, the following pages will **not** show rounded corners in IE9...

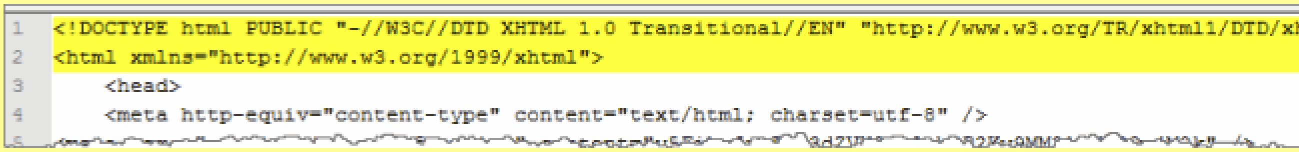


```
<html>
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

...but the following pages **will** show rounded corners in IE9:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/x
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

You can find more info on the **border-radius** capabilities here:
http://www.css3.info/preview/rounded-border/

Values: Radius X: 0 / Radius Y: 0

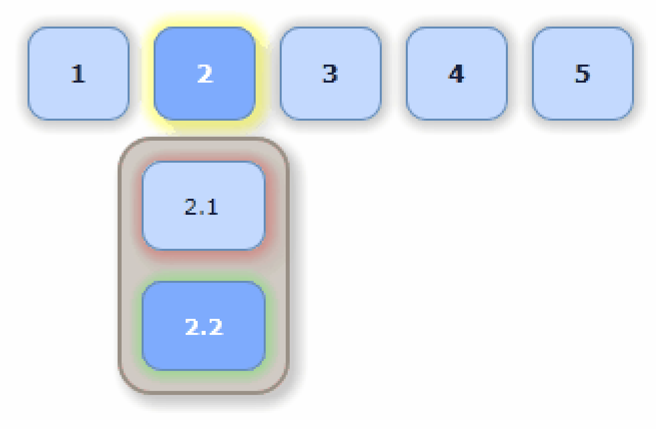
Applies to: Item

Type: Style Property

See also
Item Padding

7.4.54 Property: Item Shadow

Specifies if the item has a shadow and how this shadow looks like.

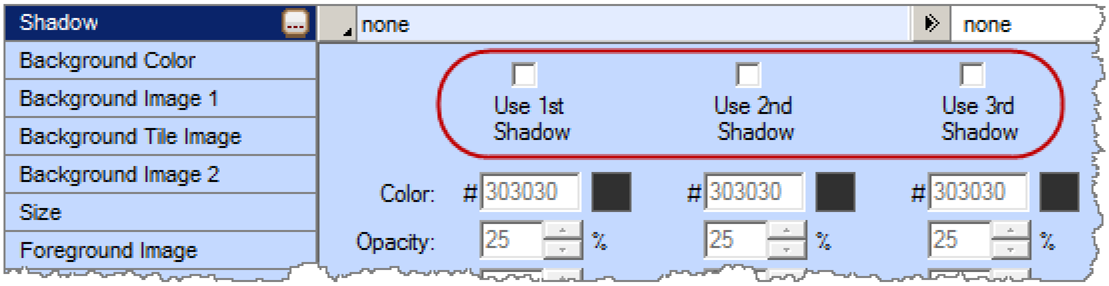


You can specify several appearance options for a shadow:

color, opacity, offsets, blur, spread

Through the styles (or style customizations) you can also use different shadows on items (see above menu).

You can mix up to **3 different shadows into a single shadow** to create unique effects:



Shadow
Background Color
Background Image 1
Background Tile Image
Background Image 2
Size
Foreground Image

none | none

□ Use 1st Shadow □ Use 2nd Shadow □ Use 3rd Shadow

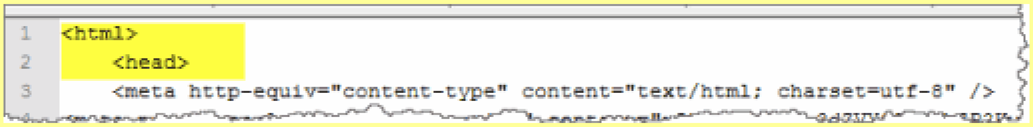
Color: #303030 #303030 #303030

Opacity: 25 % 25 % 25 %

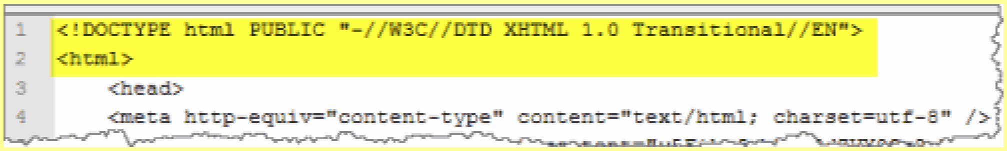
Important! The "shadow" feature is based on the **box-shadow** property of the **CSS3** specification, which is availab **Firefox 3.5+, Safari (Webkit 522+), Chrome and others**. All earlier browsers (such as IE8 or below) **do not supp** not a problem as the script degrades this feature gracefully on those browsers by **ignoring** its settings and showin shadow.

Internet Explorer 9 also requires that your pages use a **full DOCTYPE declaration**, otherwise it cannot support th property (see example below).

For example, the following pages will **not** show shadows in IE9...

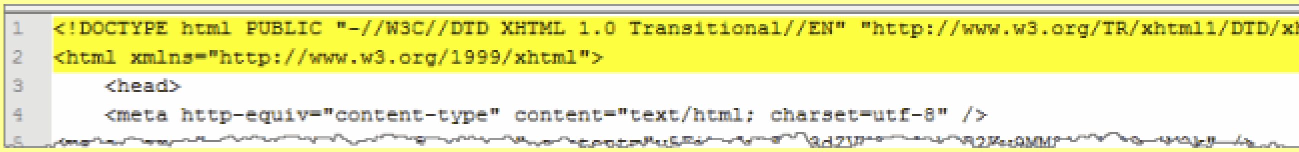


```
<html>
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

...but the following pages **will** show shadows in IE9:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/x
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

You can find more info on the **box-shadow** property capabilities here:
http://www.css3.info/preview/box-shadow/

Values: Color, Opacity, Offset X, Offset Y, Blur, Spread

Applies to: Item

Type: Style Property

See also
Group Shadow

7.4.55 Property: Item Background Color

Fills the item's background with a color.

If "none" is selected and no background image is selected (i.e. <*Item Background Tile Image*> is set to "No Image"), the group's background layer is displayed. See more on background layers.

If an <*Item Background Tile Image*> is selected, the entire background gets covered by the tiled image instead.

You can actually see the selected color along with its respective hexadecimal code. You can either enter the hexadecimal code directly or pick a color from the color dialog box which appears when clicking on the color box.

Values: #000000 (Black) - #FFFFFF (White)

Applies to: Item

Type: Style Property

See also
The different background layers

7.4.56 Property: Item Background Image 1

Shows a **fixed** image on the Item's background; LEFT-aligned relative to <Background Tile Image> or <Background Image 2>.

This image **does not tile** but occupies space based on its dimensions. You can use the "Custom Size" option to change the area that the <Background Image 1> occupies.

As shown below, the <Item Background Image 1> appears at the top-left corner of the item's background layer:

Example of using the item's background images:

Let's assume that we have the following images for our item:

<Background Image 1>: (20x 31)

<Background Image 2>: (20x 31)

<Background Tile Image>: (12x 31)

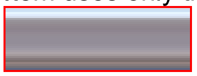
This is how the item's background shows depending on the images used (disregard the red border):

Item uses only the <Background Image 1> (left-aligned):

Item uses only the <Background Image 2> (right-aligned):

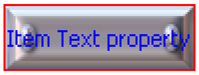
Item uses both the <Background Image 1> and the <Background Image 2>:

Item uses only the <Background Tile Image> (tiles in the entire background):

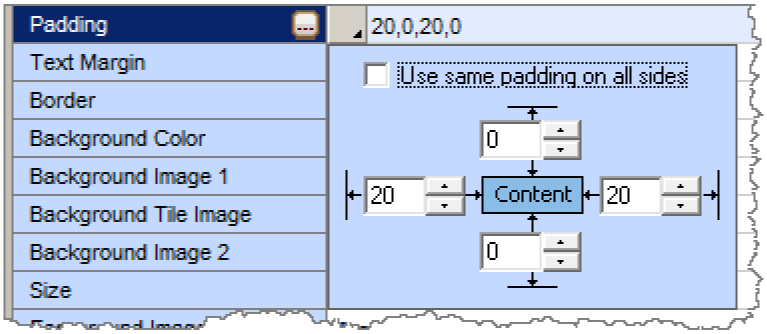


Item uses all available background images:

Note that our example uses spaces as the item's <Text>. If we put actual text there, this is what we get:



As we can see, text appears at a different layer above the background layer. We can use the <Item Padding> property to make adjustments to the content's placement:

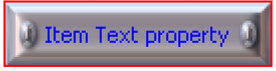


Padding: 20,0,20,0
Text Margin
Border
Background Color
Background Image 1
Background Tile Image
Background Image 2
Size

□ Use same padding on all sides

0
20 Content 20
0

By adding padding to the content (text in this example), the item appears as follows:



Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values: An image file of the .gif, .jpg or .png type

Applies to: Item

Type: Style Property

See also
The different background layers
<Item Background Image 2>

7.4.57 Property: Item Background Tile Image

Displays an image on the item's background.

This image always **tiles** on the background layer.

Example of item using <Background Tile Image>:

<Background Tile Image>: (12x 31)

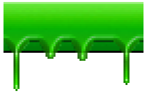
This is how the item's background shows if it uses only the <Background Tile Image> (disregard the red border):

What is the "Tiling Option":

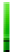
This option may be used when both the <Item Background Image 1> and <Item Background Image 2> exist on the item's background. In very rare occasions you may need to change this option from the default value to the "Tiles between them only" value.

Let's see an example in what cases this change may need to be made.

We are using the three following images on the item's background:



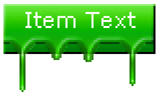
<Background Tile Image>:



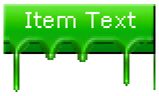
<Background Image 1>:

<Background Image 2>:

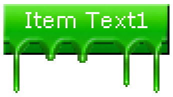
We want to get the following result:



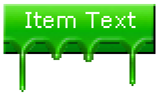
If we use the default option, the <Background Tile Image> will tile on the entire background, even below the right and left image. The result will be as follows:



As you can see, there is a line at the end as a result of the tiling of the <Background Tile Image> even below the <Background Image 2> which has transparent bottom.
To realize why this happens, view the button below which is one character bigger. You can see that the green line of the previous button is part of the melting bar that tiles to the right.



To fix this problem, you just need to check the "Tiles between them only" value on the "Tiling Option" property. The resulted button will be as follows, which is what we were looking for:



See important example at the <Item Background Image 1> property that describes how the background images can be used to deliver stylish effects.

Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values: An image file of the .gif, .jpg or .png type

Applies to: Item

Type: Style Property

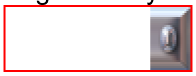
See also
The different background layers
<Item Background Image 1>

7.4.58 Property: Item Background Image 2

Shows a **fixed** image on the Item's background; RIGHT-aligned relative to <Background Tile Image> or <Background Image 1>.

This image **does not tile** but occupies space based on its dimensions. You can use the "Custom Size" option to change the area that the <Background Image 2> occupies.

As shown below, the <Item Background Image 2> appears at the top-right corner of the item's background layer:



See important example at the <Item Background Image 1> property that describes how the background images can be used to deliver stylish effects.

Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values: An image file of the .gif, .jpg or .png type

Applies to: Item

Type: Style Property

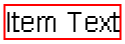
See also
The different background layers
<Item Background Image 1>

7.4.59 Property: Item Size

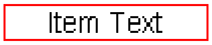
Sets a minimum width or height for the Item's background area, excluding border size.

The default "Auto" value means that the item's size is determined by its content and that it is not forced to specific values.

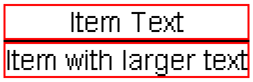
For example, the item below uses no images and no <Item Padding> or <Item Horizontal Margin>:


Item Text

If we want to enforce a larger width on this item, we can change the "Minimum Width" value to "100". The resulted item will be the following:


Item Text

Note that we refer to **minimum** width and height. This is because the actual item's size depends on other factors too, that may enlarge the minimum value. For example, the existence of a second item with larger text in the same group will cause the width to be larger than the one specified:


Item Text
Item with larger text

The borders are not counted when determining the actual item's size. This means that if the <Item Size> is "100,100" and the item uses a border with a size of 2 pixels, the actual item size will be 102,102 pixels.

Note for "CSS Menu" Genre implementations: Both the "width" and "height" of all item styles should be explicitly specified in pixels and not as "Auto". The "Auto" value is not acceptable when compiling, since a "CSS menu" cannot use javascript to calculate the size of all items on-the-fly.

Values:	"auto" or numbers in pixels
Applies to:	Item
Type:	Style Property

7.4.60 Property: Item Foreground Image

Displays an image on the Item **provided that the <Item Image> property is left blank**.

- You can use the **Image Placement** option to specify the relative positioning between the image and the item's text or choose to display the image only.

- You can use the **Image Margin** option to specify margins on the image, allowing you to choose any white space you like at its sides.

This property can be used to create styles that have bullets or arrows next to the item text. Since the <Item Foreground Image> is not a content property (like the <Item Image>), this image will be shown next to the text in all items using this style.

Important Note: The <Item Foreground Image> is not shown if the <Item Image> content property is also used.

Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values: An image file of the .gif, .jpg or .png type

Applies to: Item

Type: Style Property

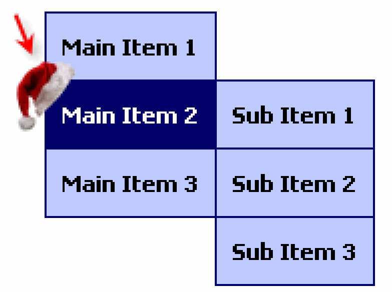
See also
Item Image

7.4.61 Property: Item Floating Image

Displays a floating image **above** the item's layer.

- The advantage of a "floating image" is that it appears above the item's layer, which means that it may also exceed the item's area.

- For example, you can simply decorate your website menu for Christmas by just adding a "floating image" on the items you want and for any state (Normal, Mouse Over or Mouse Click):



This can create interesting styles that show an image floating on the side of items.

If you want part of the image to show **outside** the item's Group (like the example shown on above screenshot), please make sure that you do the following:

- the **Group <Opacity> is 100%** (otherwise Internet Explorer does not display this part of the image).
- the **Offsets** have negative values.

Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values: An image file of the .gif, .jpg or .png type

Applies to: Item

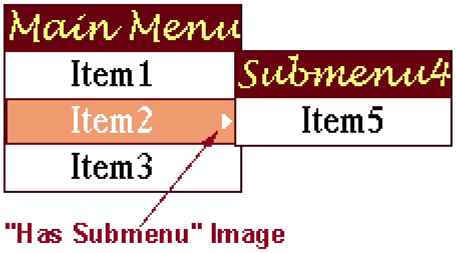
Type: Style Property

See also:
[Style Properties]

7.4.62 Property: Item "Has Submenu" Image

Adds an image to the item to emphasize the existence of a child submenu group.

The image is always positioned at the right of the Item's content.



The <"Has Submenu" Image> Appearance option lets you choose whether the image shows only when an actual submenu exists on the item or always. This option gives great flexibility when creating styles.

Note: **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

Values: An image file of the .gif, .jpg or .png type

Applies to: Item

Type: Style Property

See also:
[Style Properties]

7.4.63 Property: Item Cursor

Allows you to change the shape of the cursor when the mouse passes over the item.

Auto displays a "hand" cursor when the <Link> property is filled. In that way, it automatically shows that the item has a link towards a web page.

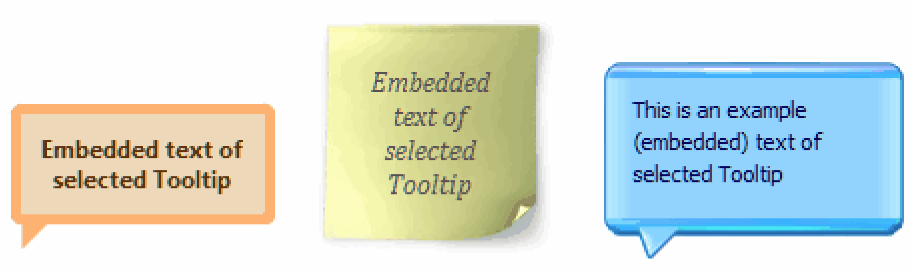
Values: a cursor selection from a list of supported cursors

Applies to: Item (on [Mouse Over] state only)

Type: Style Property

7.4.64 Property: Item Tooltip Style

Allows you to change the style of the tooltip that appears when the mouse passes over the item (replaces the browser's default style to a more attractive/usable one).

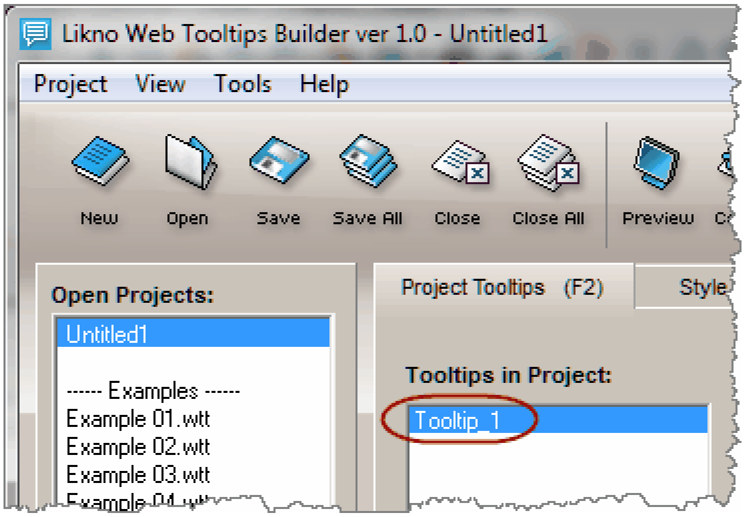

Embedded text of selected Tooltip
Embedded text of selected Tooltip
This is an example (embedded) text of selected Tooltip

The "Tooltip Style" property offers an easy integration between your menu and tooltips produced with the **separate** Likno Web Tooltips Builder product (in order to create/compile the tooltips you need to have a license of that software too).

How?

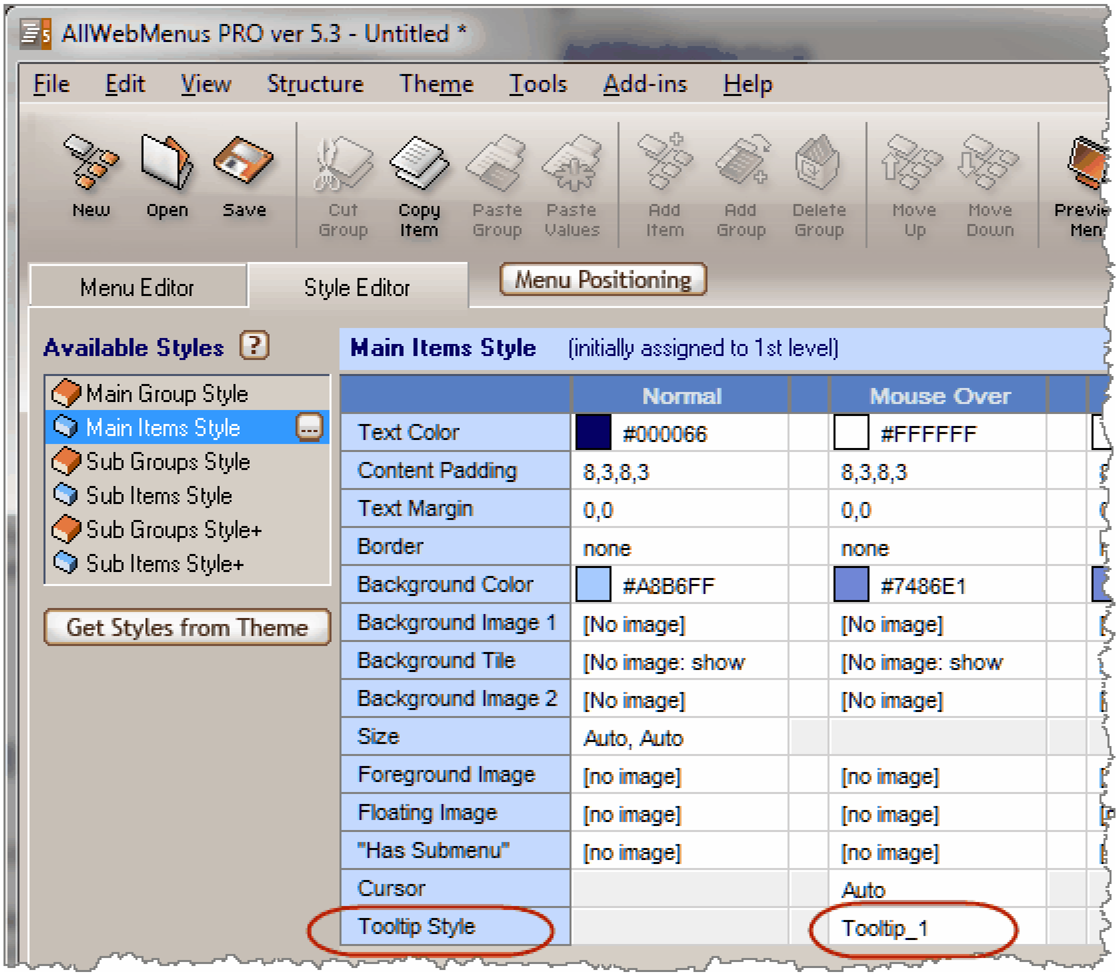
1. In **Likno Web Tooltips Builder**, compile and link your tooltip project to the page(s) where your menu appears.

2. Copy the name of your tooltip project, as shown here:



3. In **AllWebMenus** (build #810 or above), load your menu project.

4. Select the menu items or **item styles** that you want to show the custom tooltip (instead of the default one), and paste the above value to their **<Tooltip Style>** property:



Main Items Style (initially assigned to 1st level)

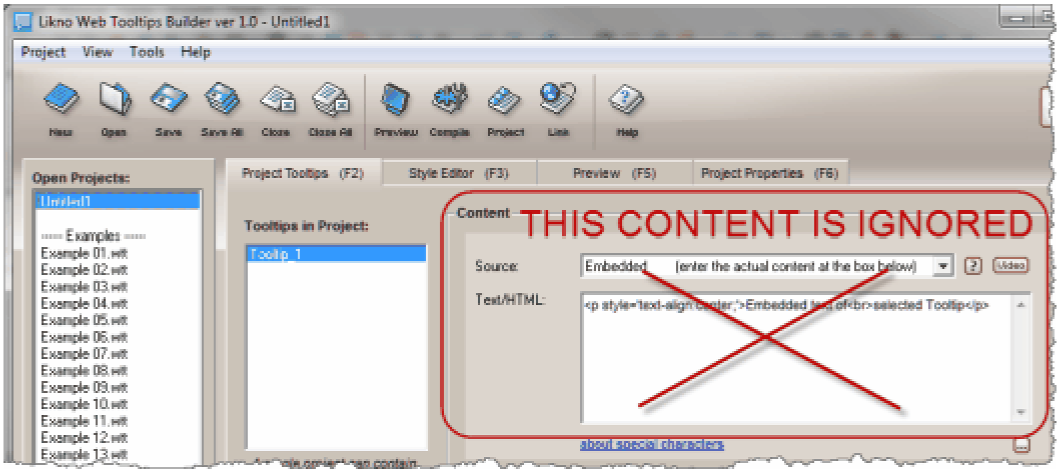
	Normal	Mouse Over
Text Color	■ #000066	□ #FFFFFF
Content Padding	8,3,8,3	8,3,8,3
Text Margin	0,0	0,0
Border	none	none
Background Color	■ #A8B6FF	■ #7486E1
Background Image 1	[No image]	[No image]
Background Tile	[No image: show	[No image: show
Background Image 2	[No image]	[No image]
Size	Auto, Auto	
Foreground Image	[no image]	[no image]
Floating Image	[no image]	[no image]
"Has Submenu"	[no image]	[no image]
Cursor		Auto
Tooltip Style		Tooltip_1

5. This is all you need to do. Just make sure that the compiled tooltip project is also linked to the web page where your menu appears. In pages where the tooltip project is not linked, the default browser tooltip is shown instead.

Example: If you want **all** your menu items to show the appearance of a custom tooltip, named "Tooltip_1", you should enter this value at the <Tooltip Style> property in **all 3 item styles**:

Main Items Style
Sub Items Style
Sub Items Style+

Note that the Likno Web Tooltips Builder project is used to change the **style** (appearance) of the menu tooltips, NOT their content.

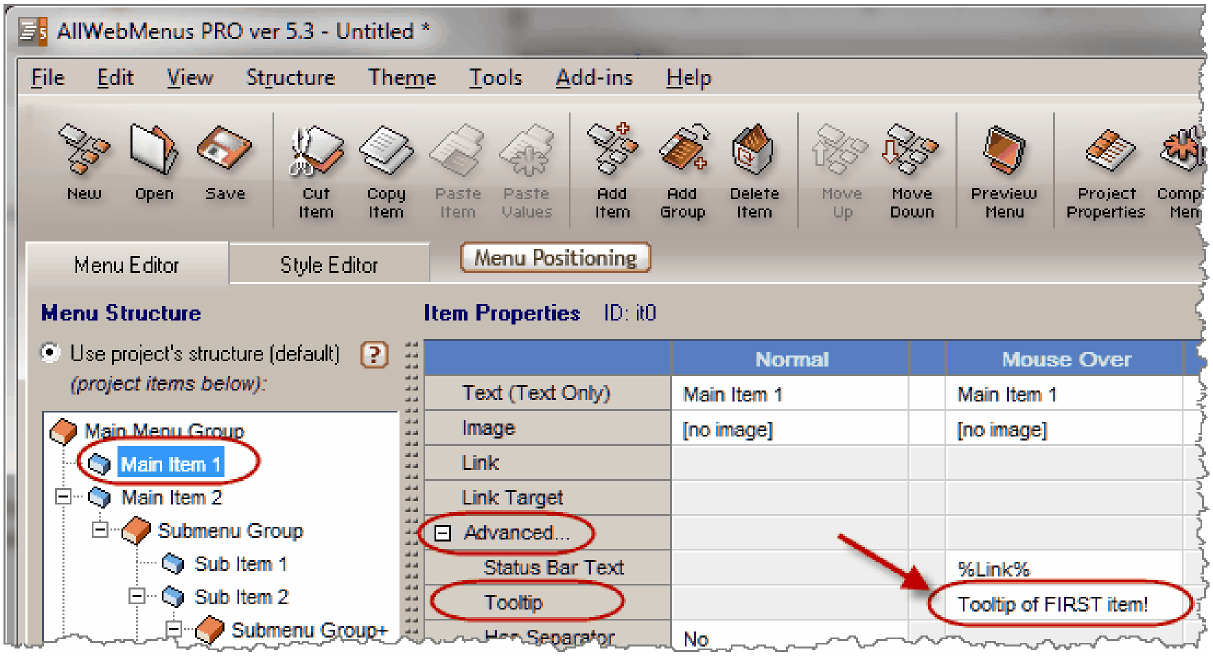


Source: Embedded (enter the actual content at the box below)

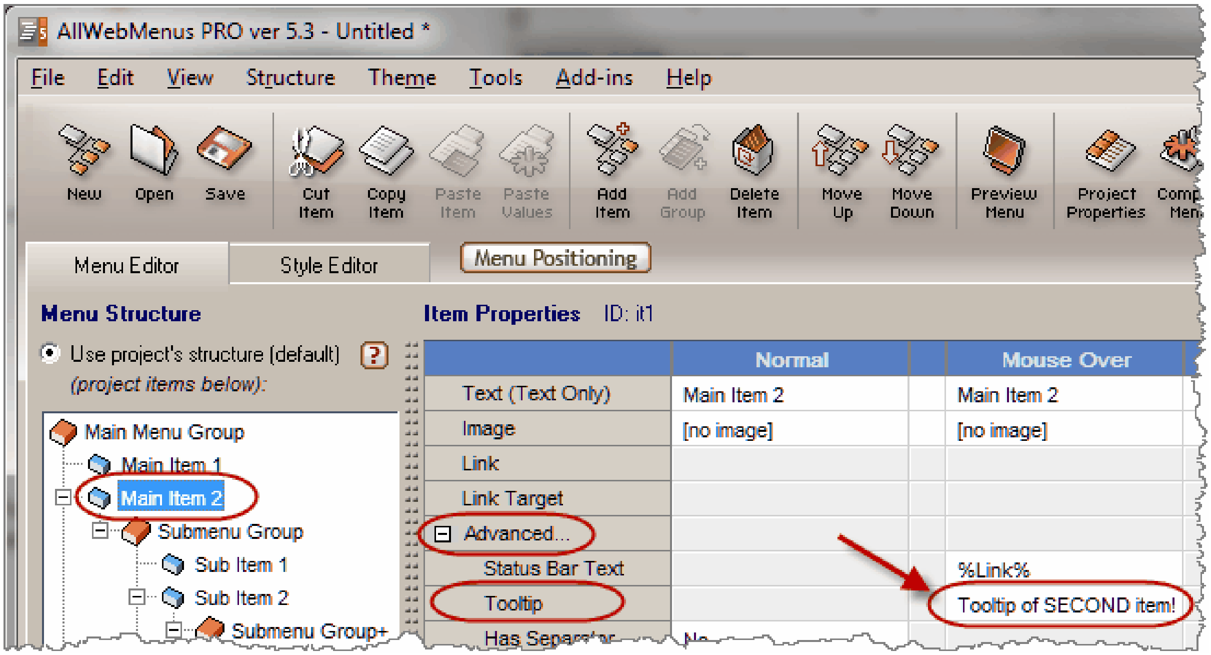
Text/HTML: <p style="text-align:center;">Embedded text of
selected Tooltip</p>

Otherwise, all tooltips of all menu items would show the same content, which is not a desirable feature. Each menu item (in AllWebMenus) has a different "Tooltip" value to show.

Example:


AllWebMenus PRO ver 5.3 - Untitled *
File Edit View Structure Theme Tools Add-ins Help
New Open Save Cut Item Copy Item Paste Item Paste Values Add Item Add Group Delete Item Move Up Move Down Preview Menu Project Properties Comp Men
Menu Editor
Style Editor
Menu Positioning
Menu Structure
Use project's structure (default)
(project items below):
Main Menu Group
Main Item 1
Main Item 2
Submenu Group
Sub Item 1
Sub Item 2
Submenu Group+
Item Properties ID: it0

	Normal	Mouse Over
Text (Text Only)	Main Item 1	Main Item 1
Image	[no image]	[no image]
Link		
Link Target		
⊟ Advanced...		
Status Bar Text		%Link%
Tooltip		Tooltip of FIRST item!
Hor Separator	No	



	Normal	Mouse Over
Text (Text Only)	Main Item 2	Main Item 2
Image	[no image]	[no image]
Link		
Link Target		
⊟ Advanced...		
Status Bar Text		%Link%
Tooltip		Tooltip of SECOND item!
Has Separator	No	

Do not forget that both your menu and tooltip projects (.js files) should be linked to the page, otherwise the menu will not be able to locate the tooltip on page and the default browser tooltip will show instead.

Notes:

- View an online example of a menu that uses tooltips

- View a video on how to use this feature!

- The tooltip style cannot be shown within the AllWebMenus "Menu Preview" pane (as this requires external files), but it will be shown on your actual pages (provided that the tooltip project exists there).

- If you use more than one tooltip in your page, make sure that you give different names to these tooltips, otherwise you will have a conflict.

Values: "Default" OR a tooltip name specified by a "Likno Web Tooltips Builder" project.

Applies to: Item (on [Mouse Over] state only)

Type: Style Property

How Do I...

Part VIII

8 How Do I...

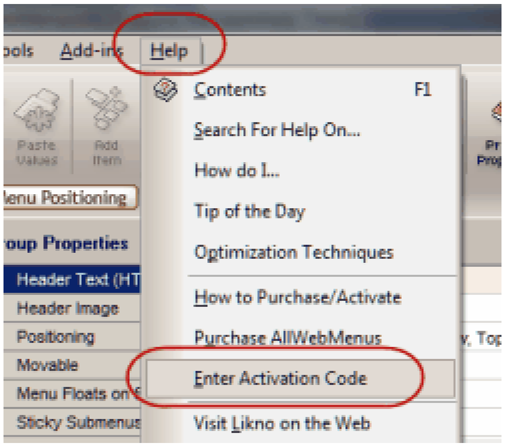
8.1 "How Do I" List

- Find my Installation ID
- Purchase/Activate
- Use AllWebMenus with my favorite Web Authoring tool
- Link menus to FrontPage Shared Borders
- Position the menu relative to a page element
- Specify which pages contain a menu, using the FrontPage interface
- Use multiple menus on the same page
- Optimize menus to improve building speed
- Create a Site Map out of my menu structure
- Replace the default right-click menu used by the browser
- Import a FrontPage Navigation Structure into AllWebMenus
- Make my submenus appear *in front* of Flash Objects, Java Applets or Select Boxes
- Make my menus appear under modal boxes, slideshows, lightbox dialogs, etc.
- Show a specific item (or submenu) as selected (highlighted) when loading the page
- Use my designed menu on my web pages
- Create a transparent menu
- Make items look like buttons
- Transfer a style to other elements
- Transfer a submenu group to another parent item
- Setup the Compile Properties of an older version awm project
- Customize a menu with "sliding" submenu groups

8.2 Find my INSTALLATION ID

In order to activate your AllWebMenus license you need to login to your Likno User Area and provide us with the unique **Installation ID** of your current installation.

Your Installation ID can be found at the **"Help** -> **Enter Activation Code"** option of the application menu, as shown below:



8.3 Purchase/Activate

The **unregistered, non-activated installation** allows you to try the entire feature set of AllWebMenus within the Preview Panel (the "Compile" and "Link" commands are disabled).

Activating AllWebMenus allows you to use your **fully-functional menus** without restrictions.

Step 1. Purchase AllWebMenus

- Access the online registration page by choosing the AllWebMenus menu option **"Help -> Purchase AllWebMenus"**

- Click on your preferred method of payment and enter the necessary info in the order form

  You will soon receive an e-mail containing instructions on how to get the Activation Code that unlocks your application.

Step 2. Activate your Installation

- Choose the AllWebMenus menu option "**Help** -> **Enter Activation Code"**
-
- Enter your Activation Code

You will need to have "Full Admin rights", read how here.

Following these simple steps you will be a **registered user** and you will be able to take **full advantage** of the AllWebMenus features.

Methods of payment
There are many flexible payment options:

- Credit Card Orders (all major credit cards accepted)
- Phone Orders (toll-free)
- Mail Orders (pay by check)
- Fax Orders
- Purchase Orders

All on-line transactions are protected by an extremely high level of encryption, the highest allowed by law.

Note:
Since version 5.1.730, we can send a special Activation Code (upon request) that allows you to use the "Compile" and "Link" commands.

This special Activation Code allows you to compile menus that can be tested under the following **conditions**:

- with **random scrambled** text on some items (instead of the actual text)
- using an **encrypted and larger file** (almost double) than the licensed one (which is optimized in size)

None of the above conditions prevents you from actually testing the entire feature set and cross-browser functionality of your menus. You will be able to realize all features at their full extend and decide whether the product fits your needs or not.

8.4 Use AllWebMenus with my favorite Web Authoring tool

AllWebMenus collaborates with all the common Web Authoring tools. In order to find out how your favorite authoring tool integrates with AllWebMenus please visit our special on-line section at http://www.likno.com/web_authoring_tools.html

Note: as "*Web Authoring tool*" we mean any tool used to create, edit and maintain web sites, such as Dreamweaver, FrontPage etc.

We have also created special integration with the following Authoring Tools:

- Dreamweaver (free Extension) more info online
- FrontPage (free Add-In) more info online
- NetObjects Fusion (free Component) more info online

8.5 Link menus to FrontPage Shared Borders

This topic is described in detail at the "Compile Properties, FrontPage Support" help topic.

8.6 Position the menu relative to a page element

This topic is described in detail at the <Positioning> help topic.

8.7 Specify which pages contain a menu, using the FrontPage interface

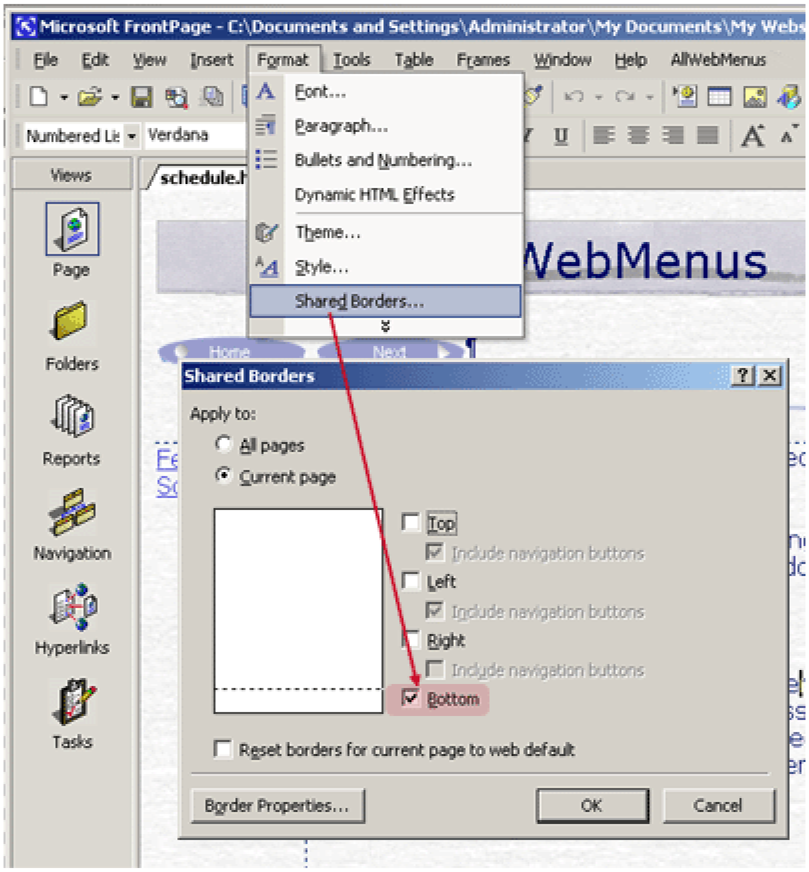
In FrontPage, the HTML code that produces the navigation menu is inserted into the Bottom Shared Border. Therefore, the navigation menu shows up in all pages containing the Bottom Shared Border.

If you are using the FrontPage Add-in, the following actions take place when pressing the "Tools -> Generate Menu" command:

1. The navigation menu is **compiled** into your Web's folder.
2. The menu is **linked** into your Web's Bottom Shared Border.
3. The first time that the menu gets generated, the Bottom Shared Border option is **added** to all pages that are **part of your Web's Navigation Structure**.

Note that action [3] takes place only the first time that a menu gets generated. In addition, action [3] does not apply the menu to **all** your pages but only to those pages that are part of your Navigation Structure.

At this point, you can manually select the **additional pages** where you want the menu to show up by simply adding a Bottom Shared Border to them too. To do so, just open these pages, select "Format -> Shared Borders" and check their Bottom Shared Border box, as shown below:



Apply to:
○ All pages
● Current page

□ Top
☑ Include navigation buttons
□ Left
☑ Include navigation buttons
□ Right
□ Include navigation buttons
☑ Bottom

□ Reset borders for current page to web default

If you want to apply the menu to all your Web pages with only one move, you can simply check the "All pages" radio box.

See also
How Do I, Position the menu relative to a page element

8.8 Use multiple menus on the same page

You can use more than one menu on the same page if you compile and link these menus with different names.

Example:
1. Compile menu1.js and link it to the page
2. Compile menu2.js and link it to the page

Two menu-linking code sections will be created inside the html of the page. One for menu1.js and a different one for menu2.js (the order is not important).

How it works:
Let's say that you want to link a menu called "menu3.js" to a page. These are the actions taken place by AllWebMenus:

Does the menu-linking code already exist for "menu3.js"?
Yes **Replace** the old menu-linking code with the new one

No Insert a **new** menu-linking code for "menu3.js" into the page

Comments:
It is obvious from the above example that an additional menu will not be added in your page if its name already exists inside that page. So, you need to be careful **not** to use the same name for different menus even if they are located in different directories.

Multiple menus can take advantage of the "Tools -> Project Properties -> Compile Properties: Folders & Files" capability so that common folders are used for the menu-building libraries and images (eg: awmdata).

You should not to use images with same names among different menus as they may override each other (unless you keep the default setting of having different subfolders for the menu images).

You can have as many menus as you like inside the same page but you should be careful not to over do it, as performance may be affected.

See also
Compile Properties: Folders & Files

8.9 Optimize menus to improve building speed

Here you can specify 2 different options:

- "**Ignore [Mouse Click] values to gain menu-building speed**" option:

There are two optimization techniques you may want to know when designing your menus:

Technique 1:
You can increase the menu-building speed if you **use the same appearance in both the [Mouse Over] and [Mouse Click] Item States**.

This is possible if you check the "**Ignore [Mouse Click] values**" option inside the **Optimization** tab, found at:

Tools-> Project Properties -> Compile Properties-> Optimization

In most menu designs, clicking on an item just opens a page inside the same window. Therefore the [Mouse Click] state will only be observed for a second or so. In these cases, checking the "Ignore Mouse Click State" box can be a very good idea.

On the other hand, if you have medium-sized menus that appear very quickly you may not want to lose the flexibility of a 3rd item state. Obviously, there is a performance/features tradeoff here and you are able to choose what fits you best.

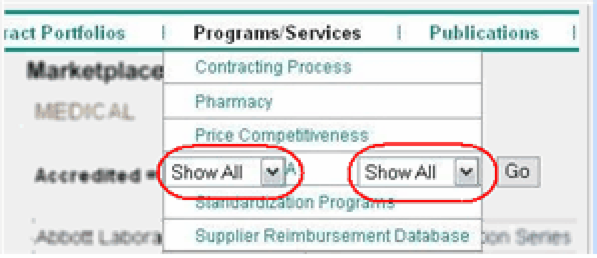
Note: If you want to use this optimization technique for your final, compiled menus, it is important that you keep the respective box checked prior to compilation.

Technique 2:
As a general rule, prefer to create more submenus with a few items (no more than 12 or 15) than create less submenus with a lot of items.

- "IE 6 fix on select boxes" option:

This is a feature that solves a bug of Internet Explorer 6 (only) when select boxes (combos) exist on page.

It appears that Internet Explorer 6 shows select boxes (combos) above submenus by default, something that does not occur on any other browser.



In that case you should check this option, which adds special code in the compiled javascript menu to cure the problem.
You have 2 options to choose from:

- The "Hide Fix", which automatically hides the select box when a submenu "intersects" its area. (recommended fix)

- The "IFRAME Fix", which uses an IFRAME to solve the problem without hiding the select box when a submenu intersects its area.
This option has two small inconveniences: It may cause a small delay in very large menus and may show a harmless "warning" if you use it on HTTPS secure websites.

You could alternatively move the combo boxes from the area where the submenus appear. If this is not feasible by your design check the related option and the problem will be fixed.

8.10 Create a Site Map from my menu structure

This topic is described in detail at the "Command: Create a Site Map" help topic.

8.11 Replace the default right-click menu used by the browser

You can replace the default right-click menu of your browser with your own menu!

To use this feature please follow the instructions below:

- First, you should not forget to set the <Positioning> property to "Manually Trigger Main Menu or any Group on user action" before you compile the menu.

- Compile the popup menu (eg: as **popup_rightclick.js**) and link it to your HTML page.

- Open your web page(s) and include the following HTML code in the HEAD section (between the <head> and </head> tags):

```
<script>
function awmShowMenu(x,y,z,v){}
document.oncontextmenu = rightclickmenu;
function rightclickmenu(e) {if (document.all)
{awmShowMenu('popup_rightclick',event.clientX+Math.max(document.body.scrollLeft,document
.documentElement.scrollLeft),event.clientY+Math.max(document.body.scrollTop,document.docu
mentElement.scrollTop));} else {awmShowMenu('popup_rightclick',e.pageX,e.pageY);}return
false;}
</script>
```

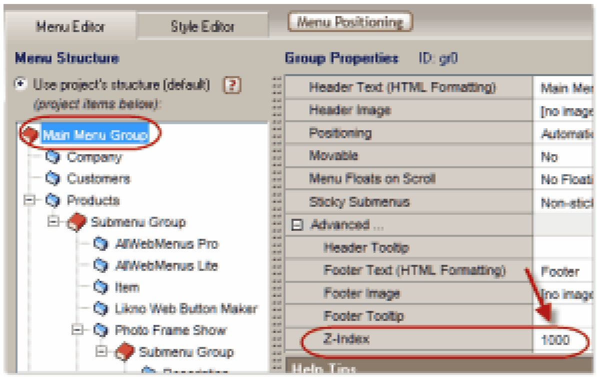
Note: change the 'popup_rightclick' name if your compiled menu is named differently.

8.12 Import a FrontPage Navigation Structure into AllWebMenus

This topic is described in detail at the "Command: Import FrontPage Navigation Structure" help topic.

8.13 Make my submenus appear *in front* of Flash Objects, Java Applets or Select Boxes

"My submenus appear under Select (Combo) Boxes, Flash Objects or Java Applets. Note that my "Z-Index" property (eg: 1000) has a larger value than the z-index value of these elements. Is there something else I can do about that?"



Group Properties ID: gr0

Header Text (HTML Formatting)	Main Me
Header Image	[no imag
Positioning	Automati
Movable	No
Menu Floats on Scroll	No Floati
Sticky Submenus	Non-stic
⊟ Advanced ...	
Header Tooltip	
Footer Text (HTML Formatting)	Footer
Footer Image	[no imag
Footer Tooltip	
Z-Index	1000

While the "Z-index property" works for most "layered" page elements, **special handling** is needed for the following elements (objects):

Flash Objects appearing above submenus

Flash objects are considered to be "windows" on top of the browser and this is why they cover the submenus by default. This can be easily changed however. The solution is to add the **transparent** parameter inside your Flash object definition, as shown on the example below:

```
<script type="text/javascript">
    AC_FL_RunContent('codebase','http://download.macromedia.com/pub/shockwave/cabs/flash
    /swflash.cab#version=7,0,19,0', 'width', '511', 'height', '250', 'src', 'ad2', 'quality', 'high',
    'pluginspage', 'http://www.macromedia.com/go/getflashplayer', 'movie', 'ad2',
    'wmode','Transparent' ); //end AC code
</script>
<noscript>
    <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version
    =7,0,19,0" width="511" height="250">
    <param name="movie" value="ad2.swf">
```

```
        <param name="WMode" value="Transparent">
        <param name="quality" value="high">
        <embed src="ad2.swf" quality="high"
        pluginspage="http://www.macromedia.com/go/getflashplayer"
        type="application/x-shockwave-flash" width="511" height="250" wmode="transparent"
        ></embed>
        </object>
    </noscript>
```

By adding the transparent parameter on the three locations above, the submenus will always appear above Flash objects on all browsers.

ALSO: Read online about the 2 distinct methods to embed Flash Player content using SWFObject (a commonly used JS library for instantiating **Flash** objects universally)

YouTube Videos appearing above submenus

YouTube videos use the "iframe" element (since February 2011), which makes a video show above submenus unless you **add** the "wmode=transparent" special parameter to its URL.

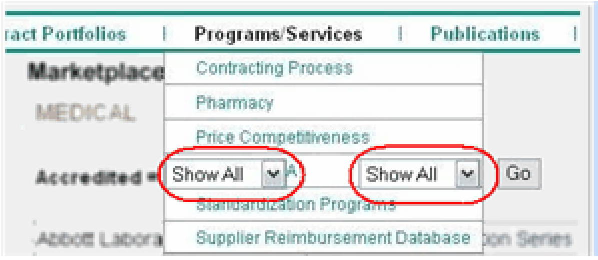
In case your YouTube URL does not use any parameters (no "?" character exists) then you should use the "?" character to add the special parameter:

```
<iframe width="560" height="349" src="http://www.youtube.com/embed/n1VTcJfL7RE
?wmode=transparent" frameborder="0" allowfullscreen></iframe>
```

In case your YouTube URL already uses parameters (the "?" character exists) then you should use the "&" character to add the special parameter:

```
<iframe width="560" height="349" src="http://www.youtube.com/embed/n1VTcJfL7RE?hd=1
&wmode=transparent" frameborder="0" allowfullscreen></iframe>
```

Select Boxes (combo boxes) appearing above submenus (IE6 problem only)



This happens in Internet Explorer 6 only (old now). The solution is given by the **"IE 6 fix on select boxes"** option at the "Project Properties -> Optimization" form.

Java applets appearing above submenus

Java applets appear as windows on top of the browser.

For Internet Explorer, you can show menus above java applets automatically if you choose the "IFRAME Fix" option at the "Project Properties -> Optimization" form.

For non-IE browsers you need to use the **awmhidediv()** and **awmshowdiv()** commands on the <Javascript Command on Appear> and <Javascript Command on Disappear> properties of the particular submenus that are covered by the applet. You need to make some additions inside your HTML for this solution to work.

The **non-IE browsers solution** has the following steps:

Step 1 - Editing the HTML page:

- Open your web page in an html editor.

- Insert the **id="awmflash1"** parameter in the <applet> tag. For example:

```
<APPLET code="Bubbles.class" id="awmflash1" width="500" height="500">
Java applet that draws animated bubbles.
</APPLET>
```

If you have more than one applet to hide, then you should use the following **strict** order for their IDs:
id="awmflash1"
id="awmflash2"
id="awmflash3"
...etc., always starting from "1", in ascending order and without any gaps.

Step 2 - Editing the .awm file:

- Open your .awm menu in AllWebMenus

- Select the submenus that are covered by the applet when they appear. Only 1st level submenus need to be selected (submenus of Main Menu items)

- Enter the following value in their <Javascript Command on Appear> property:
awmhidediv();

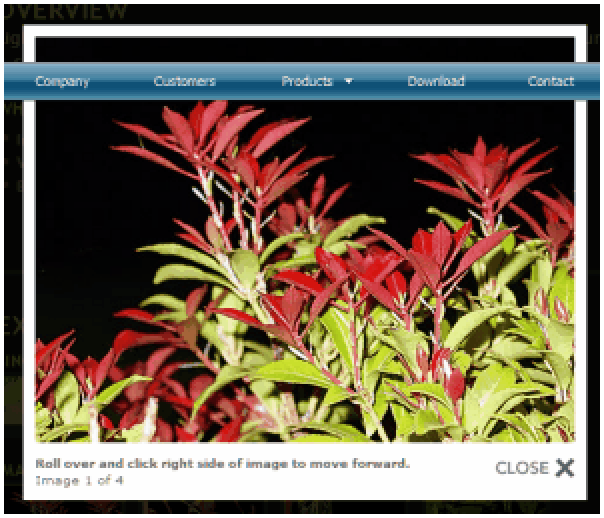
- Enter the following value in its <Javascript Command on Disappear> property:
awmshowdiv();

This solution makes the applet disappear when the submenu appears, and vice versa, so that the correct element is always on top. It is obvious that you first need to identify which submenu or submenus conflict with the applet, so that you can use the proper values in the respective Javascript Commands. Apparently, the solution works best when the menu is static, i.e. the <Menu Floats on Scroll> property is set to "No".

See also:
Make my menus appear UNDER modal boxes, slideshows, lightbox dialogs, etc. (opposite problem)

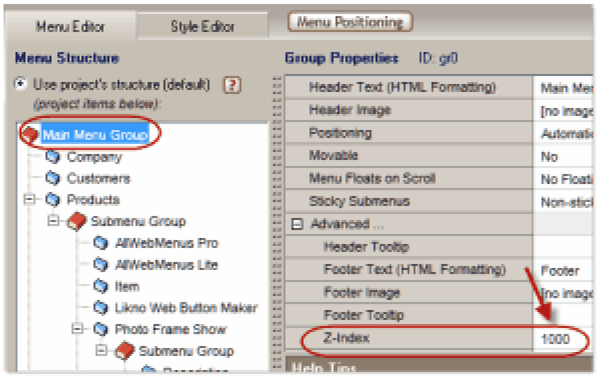
8.14 Make my menus appear *under* modal boxes, slideshows, lightbox dialogs, etc.

"My menus appear above modal boxes, slideshows, lightbox dialogs, etc. Is there something I can do about that?"



Roll over and click right side of image to move forward.
Image 1 of 4

Yes, just change the "Z-Index" property of your AllWebMenus project to a lower value than the z-index value of the modal element you are using:



Group Properties ID: gr0

Header Text (HTML Formatting)	Main Me
Header Image	[no image
Positioning	Automati
Movable	No
Menu Floats on Scroll	No Floati
Sticky Submenus	Non-stic
⊟ Advanced ...	
Header Tooltip	
Footer Text (HTML Formatting)	Footer
Footer Image	[no image
Footer Tooltip	
Z-Index	1000

If you do not know the z-index value of the modal box, just experiment with different values in AllWebMenus (like "zero"). Just make sure that changing this value does not cause your submenus appear under other page elements with higher z-index value (if any).

See also:
Make my submenus appear ABOVE Select Boxes and Flash/Applet Objects (opposite problem)

8.15 Show a specific item (or submenu) as selected (highlighted) when loading the page

There are 3 options to achieve this:

Option 1 (automatically, based on URL, for item highlighting only)

Option 2 (manually, standard item/group, for both item and group highlighting)

Option 3 (manually, custom item/group per page, for both item and group highlighting)

Read the help section of the <Show Item or Group on Appear> property on how to use these options.

8.16 Use my designed menu on my web pages

To use a menu in your web pages you must take the following four steps:

Step 1. **Compile** the menu
Step 2. **Link** the compiled menu to your web pages
Step 3. **Upload** the necessary menu files
Step 4. **Upload** the web pages that contain the menu link

The previous steps are needed for the creation of a **new menu** which is **not linked** to your web pages. Not all steps are needed for the following common objectives:

Situation
Menu is already compiled and uploaded
Web pages are already linked to the compiled menu and uploaded

Objective 1
Update the menu (apply a different appearance or add new items)
Steps needed
1 & 3
Steps not needed
2 & 4 (because the pages already contain the link and are uploaded)

Objective 2
Add the menu link to additional web pages (eg: newly created ones)
Steps needed
2 & 4 to the new pages only
Steps not needed
1 & 3 (because the menu remains the same)

See also
Compile Menu
Link Compiled Menu to Web Page(s)

8.17 Create a transparent menu

Set all background colors to "None".
Set all background images to "No Image".

Optional: Set all border styles to "none".

8.18 Make items look like buttons

Use a different Border style/width in the different item states.
Eg: Use "solid" in **Normal** item state and "inset" in **Mouse Over** item state.

8.19 Transfer a style to other elements

1. Select the element (group or item) with your preferred style
2. Copy the element (**Ctrl+C**)
3. Select the element to which the style properties should be transferred
4. Paste all style values there (**Ctrl+A**)

Repeat steps 3-4 to "paint" more elements with your preferred style.

8.20 Transfer a submenu group to another parent item

1. Select the submenu group
2. **Cut** the submenu group (this applies to its children too)
3. Select the new parent item
4. **Paste** the submenu group

8.21 Setup the Compile Properties of an older version awm project

We advice that you read the Compile Properties -> Folders & Files help section before reading this section so that you can understand the concepts easier here.

This section will help you understand how a v3.x (or older) project is automatically migrated to version 4.

Let's assume that you open a v3.x project that was compiled as:

C:\My Site Root\mymenu\navmenu.js

The project has the following properties stored within:

Libraries Folder: awmData-navmenu (default for navmenu)
Images Folder: awmData-navmenu (default for navmenu)

When opening this project with version 4, the migration process populates the related properties with the following values:

Site_Root folder: C:\My Site Root
Compiled menu name: navmenu *(custom)*

In Advanced Options:
Compiled menu subfolder: mymenu *(custom)*
Menu libraries subfolder: awmData-navmenu *(custom)*
Menu images subfolder: awmData-navmenu *(custom)*

In v3.x and below, the **Compiled menu name** was specified every time the "Compile Menu" command was executed. In version 4, the Compiled menu name is part of the Compile Properties and defaults to "**menu**". When opening an older .awm project with version 4, AllWebMenus finds the compiled menu name which is stored in the project and uses this automatically as a "custom" Compiled menu name. The same applies if the compiled menu is located at a subfolder of the site root. In this case, the **Compiled menu subfolder** property is also getting a "custom" value that shows this subfolder.

In v3.x and below the **Site_Root folder** was not specified. In version 4 this property is very important as it is used to assess the relative path of the pages specified in the <Link> property. Be careful to specify the correct Site_Root folder, as AllWebMenus will automatically convert all <Link> properties so that they take into account the relative path to the site root (Site_Root folder).

See also:
Compile Properties -> Folders & Files

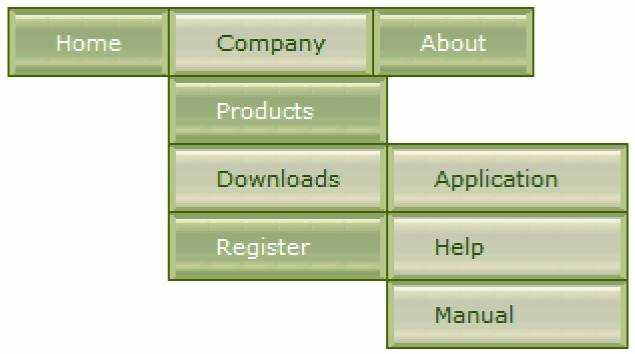
8.22 Customize a menu with "sliding" submenu groups

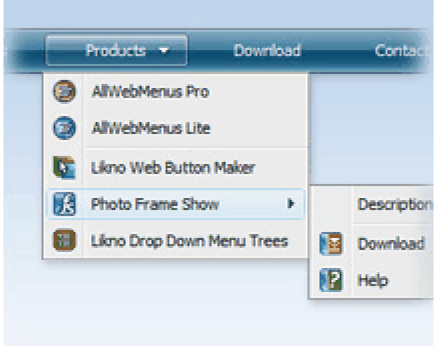
Note: *You need to purchase the "Sliding Menu Add-in" in order to use this additional feature to AllWebMenus.*

A submenu group can appear either as "drop-down" (non-sliding) or as "sliding".

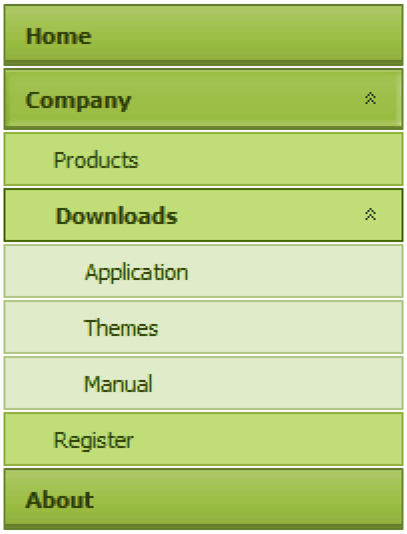
A sliding submenu group pushes the items of its parent group downwards when opening. In the same manner, it pulls these items upwards when it closes.

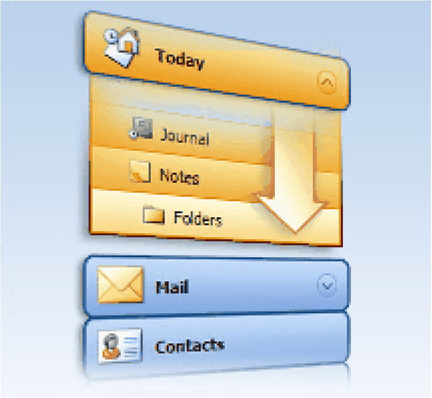
Menu with **Drop-Down** submenus (default):





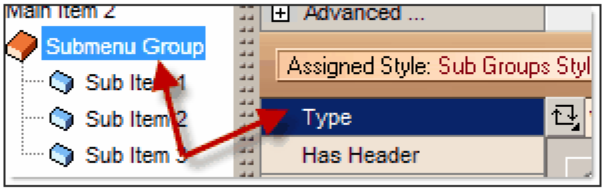
Menu with **Sliding** submenus:



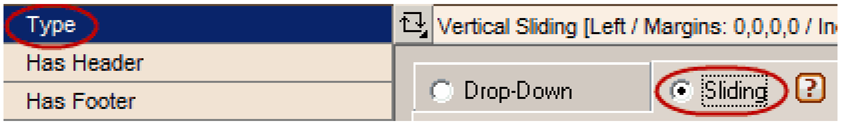


How to setup a sliding submenu:

Choose a **submenu** group or style (note: Main Menu cannot be "sliding" itself) and select its <Type> property:



Click the property button, make it "vertical" and select the "Sliding" option:



Edit the related options to further customize the sliding behavior of the selected group (or style) as you wish and you are done.
You may also browse for sliding menu themes at Theme Pack 37 and above.

Important notes on the "sliding" feature:

- A vertical group behaves as **sliding** only if the **parent group is also vertical** (otherwise the vertical "sliding" effect has no meaning). If the **parent group is horizontal** then the group behaves as **non-sliding** (drop-down), even if its "sliding" option is checked.

- The "more options" settings (i.e. multiple columns, scrollbar) are not effective if at least one item of the group has a sliding submenu.

- Do not use a <Link> (URL) in the child item, otherwise the linked URL will open instead of the sliding submenu.

- The following group properties are affected when the "Sliding" option is selected on a submenu group:

Appears - Where & Appears - Aligned with

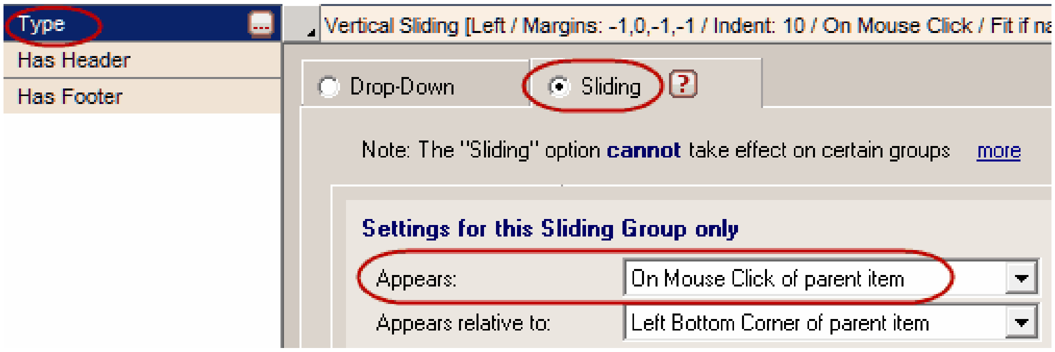
The values of the above properties are completely ignored by the sliding submenu groups (but they still affect any existing "drop-down" submenus).

Instead, the sliding submenu groups use their own properties for their appearance inside their "Sliding" tab (see image below).

Appears - How & Disappears - How

The values in the above properties are also ignored by the sliding submenu groups, except of the "Special Effects" values that are effective on both "sliding" and "drop-down" submenus.

Note: Choose either "mouse click" or "mouse over" through the **"Appears"** field inside the Type's "Sliding" tab instead.



Type
Has Header
Has Footer

Vertical Sliding [Left / Margins: -1,0,-1,-1 / Indent: 10 / On Mouse Click / Fit if na

○ Drop-Down ● Sliding ?

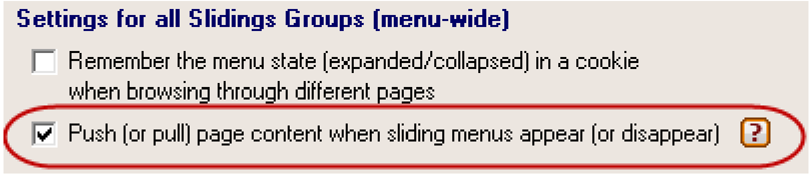
Note: The "Sliding" option **cannot** take effect on certain groups more

Settings for this Sliding Group only

Appears: On Mouse Click of parent item

Appears relative to: Left Bottom Corner of parent item

- Concerning the **"Push (or pull) page content"** option:

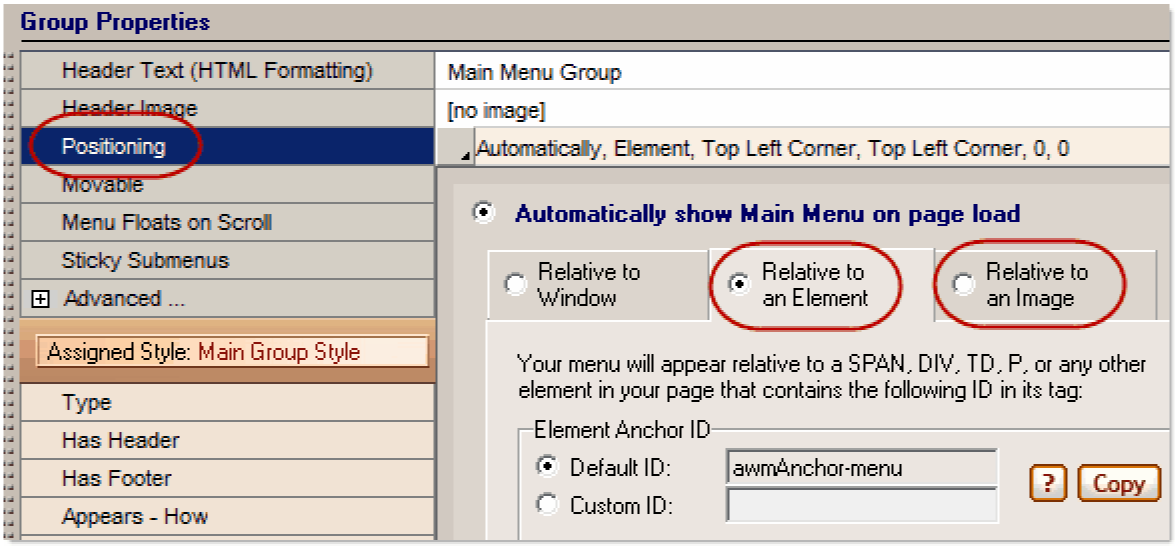


Settings for all Slidings Groups (menu-wide)

☐ Remember the menu state (expanded/collapsed) in a cookie when browsing through different pages

☑ Push (or pull) page content when sliding menus appear (or disappear) ?

This option is effective only when you position your menu relative to an Element or Image (**not** Window):



Group Properties

Header Text (HTML Formatting)	Main Menu Group
Header Image	[no image]
Positioning	Automatically, Element, Top Left Corner, Top Left Corner, 0, 0
Movable	
Menu Floats on Scroll	
Sticky Submenus	
⊞ Advanced ...	
Assigned Style: Main Group Style	
Type	
Has Header	
Has Footer	
Appears - How	

● **Automatically show Main Menu on page load**

○ Relative to Window ● Relative to an Element ○ Relative to an Image

Your menu will appear relative to a SPAN, DIV, TD, P, or any other element in your page that contains the following ID in its tag:

Element Anchor ID

● Default ID: awmAnchor-menu

○ Custom ID:

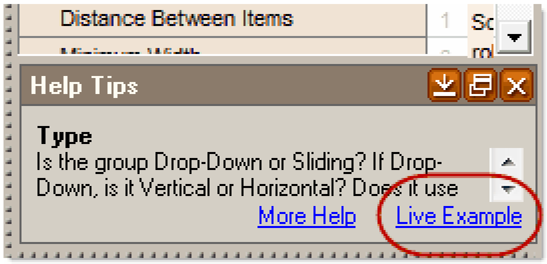
? Copy

Every time the Main Menu expands (due to its sliding submenus), the positioning element that is associated with the menu (through the Anchor ID) will increase its height accordingly. In that manner the menu "pushes" the page content. When a sliding submenu closes, the height of the element decreases accordingly so that the Main Menu "contracts" and the page content is pulled back.

If you want to use this option, it is better that you use an invisible element for positioning (empty DIV, SPAN, 1x1 transparent image etc.) so that it will not break the appearance of the page when increasing/decreasing its height. The same applies when using an image for positioning.

Notes:
- Avoid using a <TD> as the attached element of a sliding menu. Use a SPAN or DIV inside the <TD> instead.
- If the element is within a <TD>, you may want to add the ***valign="top"*** to the parent <TD> tag so that you do not have cross-browser issues.

You may also find an **example** on how this works at the "Live Example" link of the **<Type>** property.



Distance Between Items 1 Sc

Help Tips

Type
Is the group Drop-Down or Sliding? If Drop-Down, is it Vertical or Horizontal? Does it use

More Help Live Example

- Concerning the **"Ignore the <Link> value of parent items"** option:

This option is by default checked.

It is useful when you want to show a sliding submenu when **clicking** on its parent item, but this item has a page link on its <Link> property.

If you do not ignore the <Link> value, the **"click"** action will result in opening the new page (and therefore moving away from your current page).

If you wish to have this result (combining it with the "Remember the menu state" option), then uncheck this option.

See also:
Type
Available Styles

What is...

Part IX

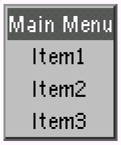
9 What is...

9.1 The three Item States

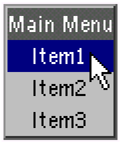
The items may look different when certain events occur. You can therefore use a different design to emphasize the event of an item getting the focus or the event of the mouse clicking on the item.

This is implemented by applying a different value to the properties for each one of the following states:

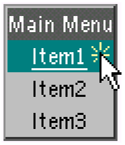
Normal state
Specifies how the item looks when the **mouse is away** from the item.
See Item1 on Normal state:



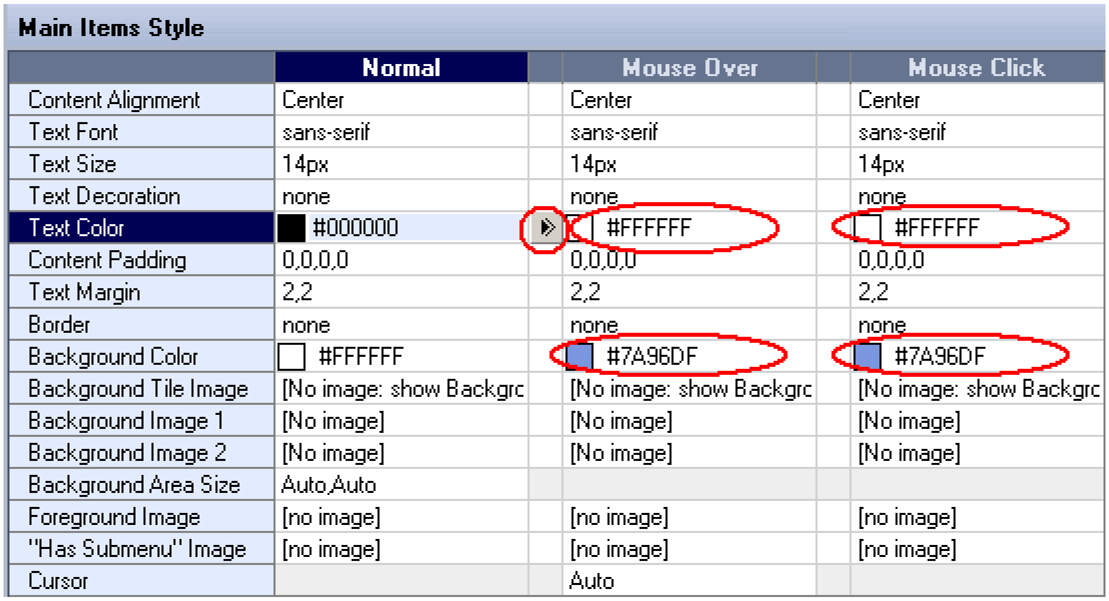
Mouse Over state
Specifies how the item looks when the **mouse is over** the item.
See Item1 on Mouse Over State:



Mouse Click state
Specifies how the item looks when the **mouse clicks on** the item.
See Item1 on Mouse Click State:



Now, take a look at the Item Properties to see how the different design of each state was actually implemented:



Main Items Style

	Normal	Mouse Over	Mouse Click
Content Alignment	Center	Center	Center
Text Font	sans-serif	sans-serif	sans-serif
Text Size	14px	14px	14px
Text Decoration	none	none	none
Text Color	■ #000000	▶ □ #FFFFFF	□ #FFFFFF
Content Padding	0,0,0,0	0,0,0,0	0,0,0,0
Text Margin	2,2	2,2	2,2
Border	none	none	none
Background Color	□ #FFFFFF	🟦 #7A96DF	🟦 #7A96DF
Background Tile Image	[No image: show Backgrc	[No image: show Backgrc	[No image: show Backgrc
Background Image 1	[No image]	[No image]	[No image]
Background Image 2	[No image]	[No image]	[No image]
Background Area Size	Auto,Auto		
Foreground Image	[no image]	[no image]	[no image]
"Has Submenu" Image	[no image]	[no image]	[no image]
Cursor		Auto	

You may have noticed that most properties keep the same value in all 3 (or just 2) item states. When editing a value in a state you can use the ▶ button to also copy this value to the states (or state) next to it.

The "Normal" state value can be copied to the "Mouse Over" and "Mouse Click" states for the selected property. Similarly, the "Mouse Over" value can be copied to the "Mouse Click" state.

9.2 The two kinds of Alignment

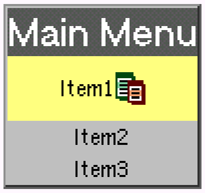
There are two kinds of alignment: The **Content Alignment** and the **Image Placement** alignment.

Content consists of both text and image, if any. This combined content is aligned on the item using the <Item Content Alignment> property.

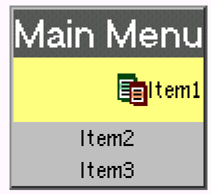
"Image Placement" is an option of the <Item Image> property and it is used to specify the position of the image relative to the text.

Examples:

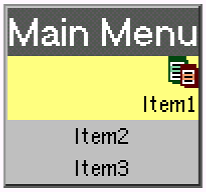
Content Alignment:	Center
Image Placement:	Right of Text



Content Alignment:	Right
Image Placement:	Left of Text



Content Alignment:	Right
Image Placement:	Above Text



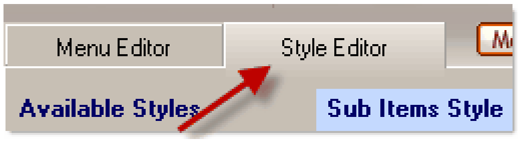
See also
Item Content Alignment
Item Image

9.3 The Assigned Style and Available Styles

Assigned Style

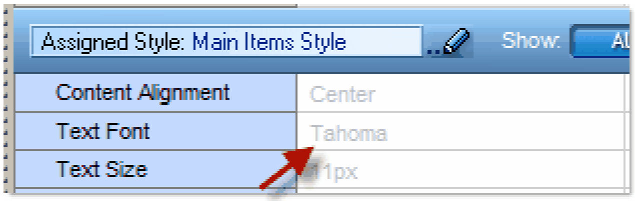
"Assigned" style means that an item (or group) that uses this style shares the same appearance/behavior with all other items (or groups) assigned to the same style.

Therefore, by using styles you can quickly edit the appearance and behavior of your menu from a central location: the **Style Editor Tab**:



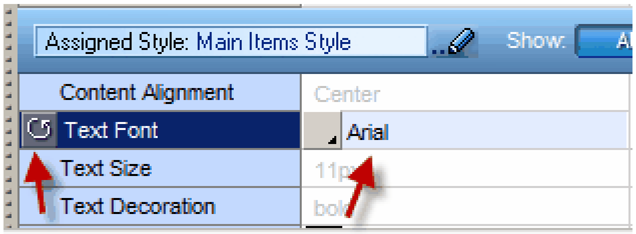
You may also "customize" a style property for any item (or group) so that it has a different value than its Assigned Style value. In this case, any changes you make to the item's (or group's) Assigned Style (through the Style Editor) will not affect the customized property, which always retains its customized value.

Non-customized property:



Content Alignment	Center
Text Font	Tahoma
Text Size	11px

Customized property:



Assigned Style: Main Items Style

Show: A

Content Alignment	Center
Text Font	Arial
Text Size	11p
Text Decoration	bol

Available Styles

Your menu uses 3 different styles for its groups and 3 different styles for its items:

🟥 **Main Group Style** (initially assigned to the **Main Menu Group,** i.e. 1st level group)

◻ **Main Items Style** (initially assigned to the **items** of the Main Menu Group, i.e. 1st level items)

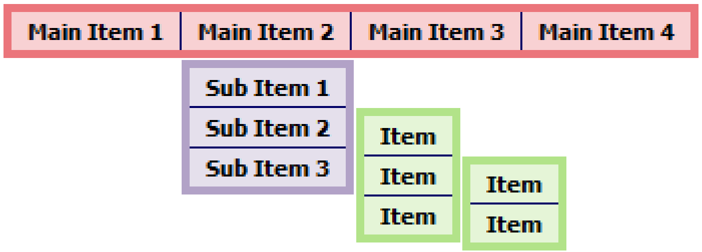
🟪 **Sub Groups Style** (initially assigned to the **1st submenus** stemming from the Main Menu items, i.e. 2nd level groups)

◻ **Sub Items Style** (initially assigned to the **1st submenu items** contained in the above submenus, i.e. 2nd level items)

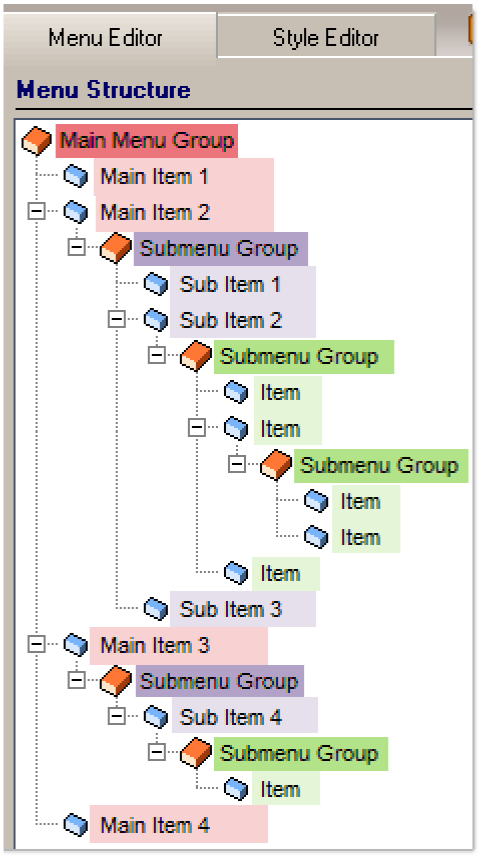
🟩 **Sub Groups Style+** (initially assigned to the **rest of submenus** stemming from the above items, i.e. 3rd level groups and beyond)

◻ **Sub Items Style+** (initially assigned to the **rest of submenu items** contained in the above submenus, i.e. 3rd level items and beyond)

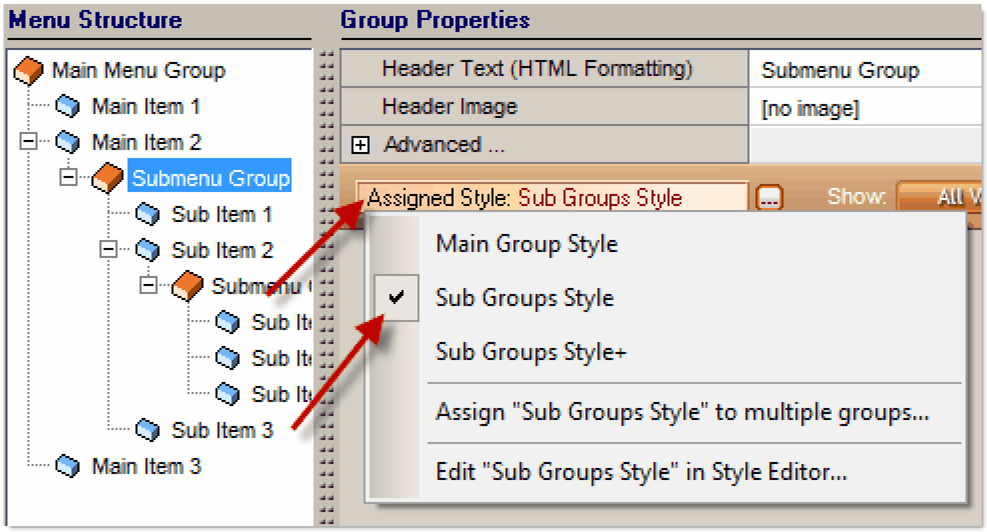
This is how a menu would look if the <Background Color> and <Border> on every item or group had the above colors that demonstrate the initial style assignments:



Initial assignments of styles:



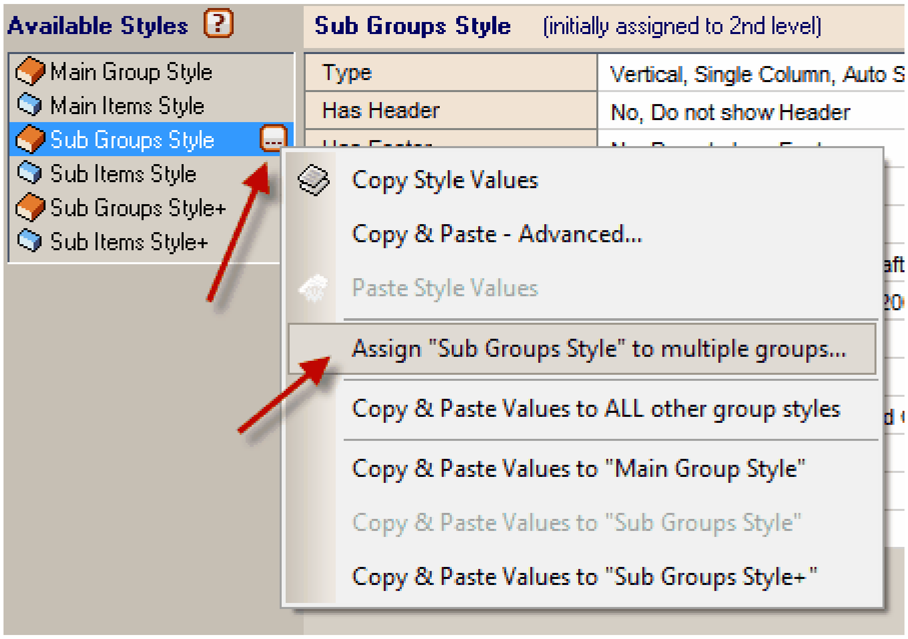
Note: The above assignments are the **initial** ones.
If you wish, you can later change the assignments by clicking on the "Assigned Style" area through the Menu Editor:



Header Text (HTML Formatting)	Submenu Group
Header Image	[no image]
⊞ Advanced ...	

Assigned Style: Sub Groups Style

or by clicking on the "..." button through the Style Editor:

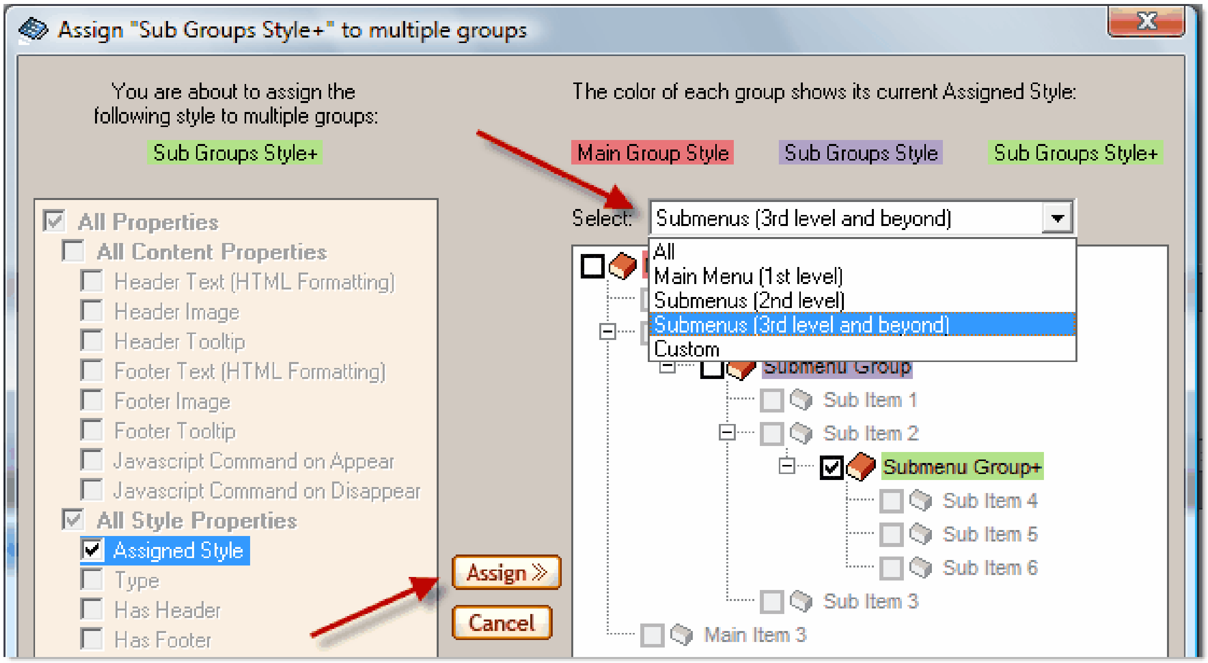

Available Styles ?
Main Group Style
Main Items Style
Sub Groups Style
Sub Items Style
Sub Groups Style+
Sub Items Style+
Sub Groups Style (initially assigned to 2nd level)
Type
Vertical, Single Column, Auto S
Has Header
No, Do not show Header
Copy Style Values
Copy & Paste - Advanced...
Paste Style Values
Assign "Sub Groups Style" to multiple groups...
Copy & Paste Values to ALL other group styles
Copy & Paste Values to "Main Group Style"
Copy & Paste Values to "Sub Groups Style"
Copy & Paste Values to "Sub Groups Style+"

Notes on automatic assignments:

- By default, when you create a new submenu it is assigned to the "Sub Groups Style" if it is on the 2nd level or to the "Sub Groups Style+" if it is on the 3rd level and beyond. This also applies to its new items ("Sub Items Style" if on 2nd level, "Sub Items Style+" if on 3rd level and beyond).

- When you apply a Theme to a menu all its submenus and items get the default assignments.

- Existing menus created with build 762 or below (i.e. before the "Style+" was introduced) will retain their "Sub Groups Style" assignment on all their existing submenus (even the ones at 3rd level and beyond). Only the **new** submenus that you create from 3rd level and beyond will be assigned to the "Sub Groups Style+". If you wish, you can re-assign all existing assignments of old projects to the default ones using the "Select" option:



Assign "Sub Groups Style+" to multiple groups

You are about to assign the following style to multiple groups:

Sub Groups Style+

The color of each group shows its current Assigned Style:

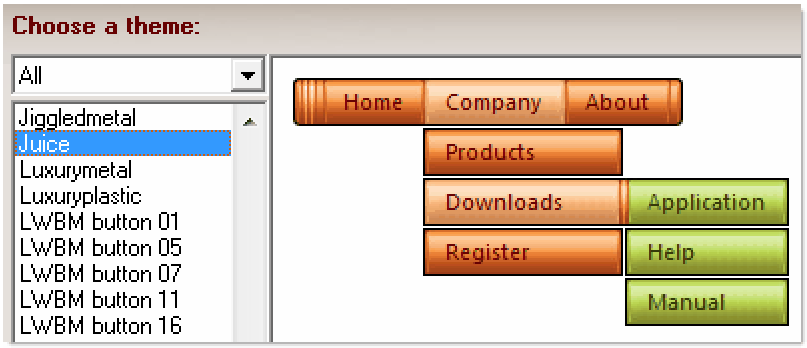
🟥 Main Group Style 🟪 Sub Groups Style 🟩 Sub Groups Style+

Select: Submenus (3rd level and beyond)

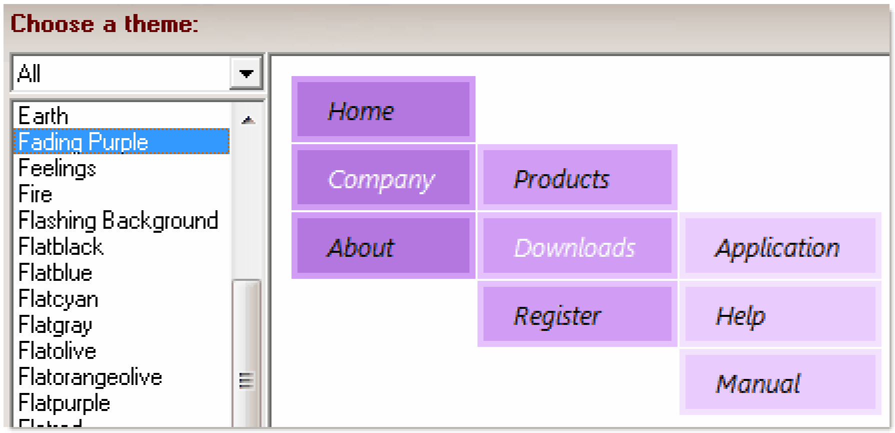
- ☑ All Properties
  - □ All Content Properties
    - □ Header Text (HTML Formatting)
    - □ Header Image
    - □ Header Tooltip
    - □ Footer Text (HTML Formatting)
    - □ Footer Image
    - □ Footer Tooltip
    - □ Javascript Command on Appear
    - □ Javascript Command on Disappear
  - ☑ All Style Properties
    - ☑ Assigned Style
    - □ Type
    - □ Has Header
    - □ Has Footer

Why is the 3rd style (i.e. "Style+") useful?

1. You can use themes that have different appearance on 3rd-level submenus and beyond:

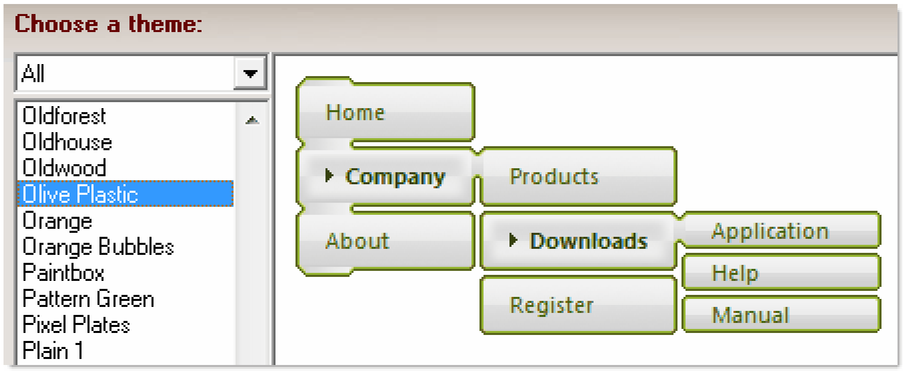


Choose a theme:

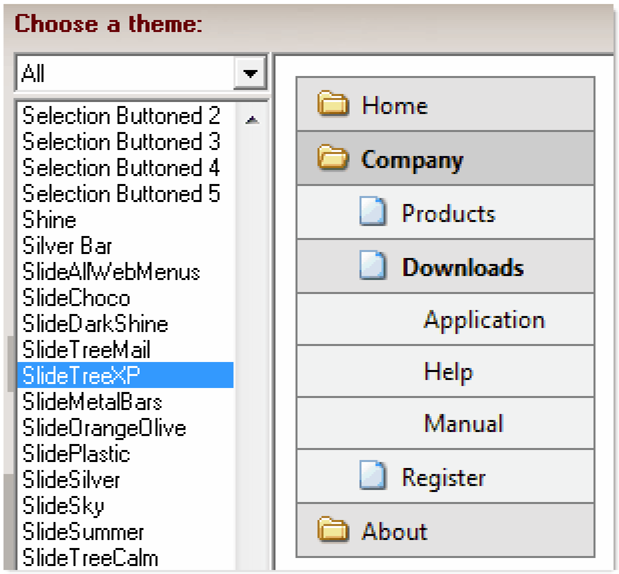


Themes that use this technique: *Juice, Fading Purple, FlatSky, Orange Bubbles (free, contained in your AllWebMenus by default)*
You may also find extra themes using this technique from **Theme Pack 36** and above at: http://www.likno.com/awmthemes.php

For example, at theme "Olive Plastic" (Theme Pack 36), its 3rd-level items (and beyond) use less padding than its 2nd-level items:



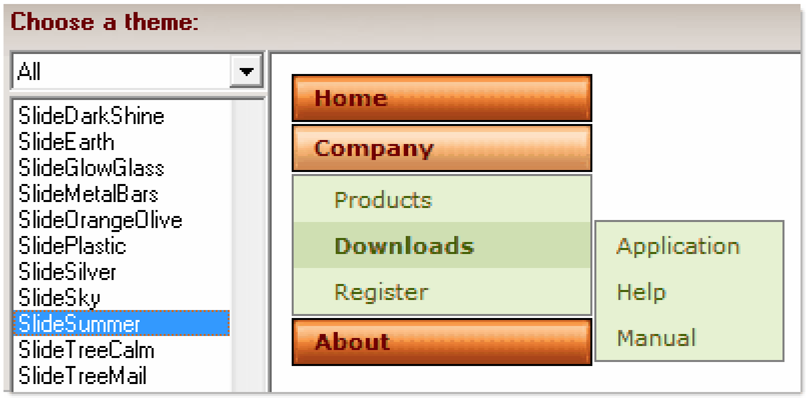
At the following **sliding menu** theme (Theme Pack 38, "Sliding Menu Add-in" required), the 3rd-level items (and beyond) do not use an icon, while the 2nd-level items do:



Such arrangements would not have been possible if all submenus had the same style.

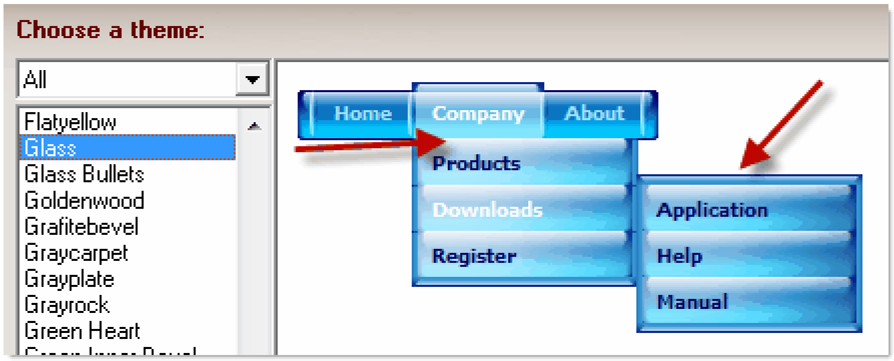
2. You can apply very interesting behaviors to **sliding menus** (available through the extra "Sliding Menu Add-in"), that combine both "sliding" and "non-sliding" (drop-down) submenus!

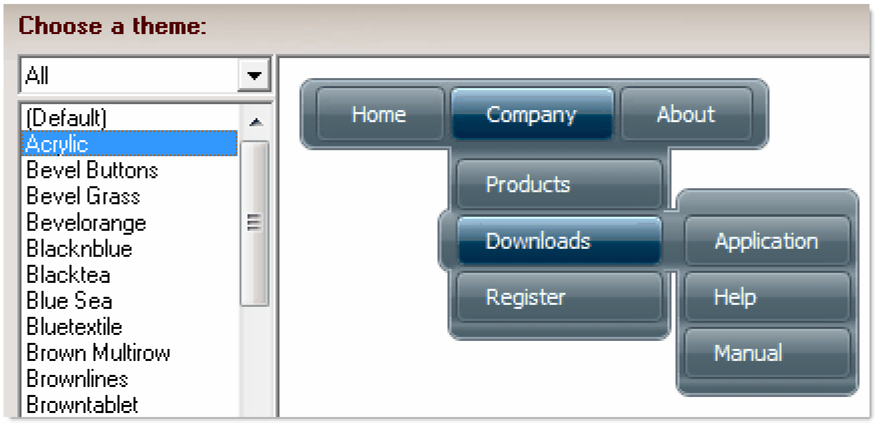
For example, the **SlideSummer theme** below (Theme Pack 37, "Sliding Menu Add-in" required) uses a sliding submenu for all its Main Menu items, while the rest of its submenus are the standard non-sliding ones. Such "sliding/non-sliding" menu combinations are very suitable in many website layouts and would not have been possible if all submenus had the same style.



3. The 3rd-level submenus can differ from the 2nd-level submenus by only a small but very important detail, that allows for a different design if the parent group is horizontal or vertical.

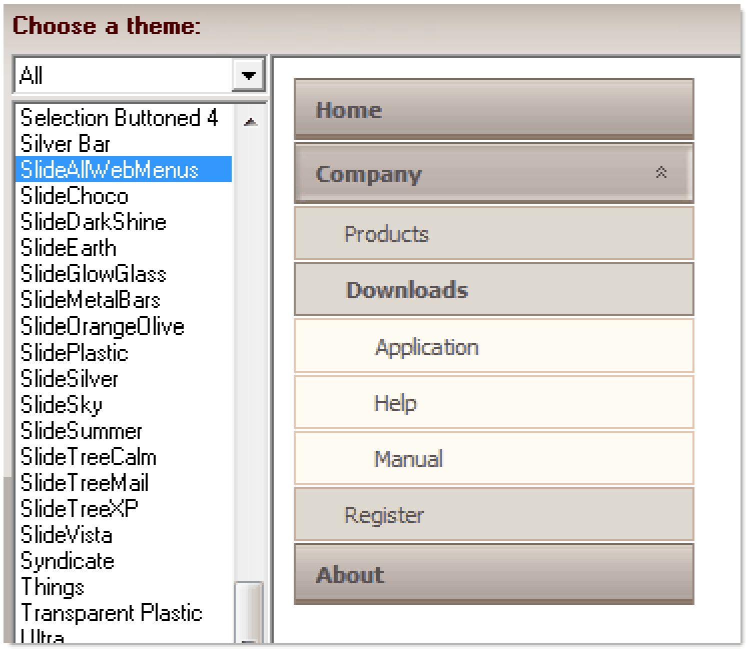
At the following examples the 2nd-level submenu does not have a "Header" (its <Has Header> property is set to "No"), while the 3rd-level submenu (and beyond) needs to have a "Header" so that the design closes properly at top:



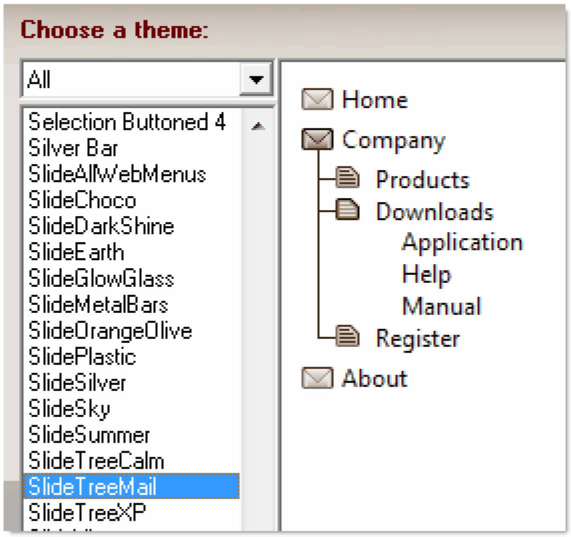


At the following **sliding menu themes** (Theme Pack 38, "Sliding Menu Add-in" required) the 2nd-level submenus are different than the 3rd-level submenus, either in colors, images or indentation:

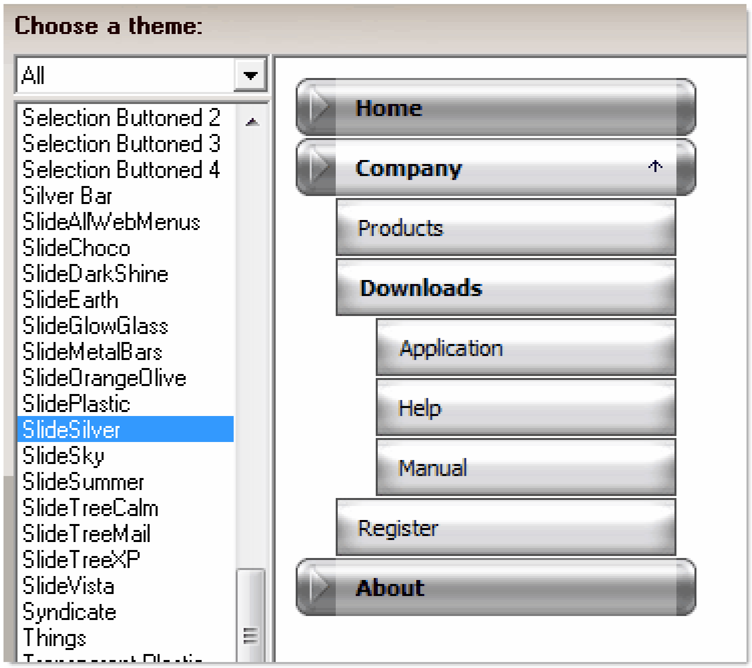
Different colors & indentation:



Different icons & indentation:



Different indentation:


Choose a theme:

All

- Selection Buttoned 2
- Selection Buttoned 3
- Selection Buttoned 4
- Silver Bar
- SlideAllWebMenus
- SlideChoco
- SlideDarkShine
- SlideEarth
- SlideGlowGlass
- SlideMetalBars
- SlideOrangeOlive
- SlidePlastic
- SlideSilver
- SlideSky
- SlideSummer
- SlideTreeCalm
- SlideTreeMail
- SlideTreeXP
- SlideVista
- Syndicate
- Things

9.4 The Image Manager

The Image Manager contains all images used in your menu along with template images.

Note: For Help on a specific Image property please click on the related link below:

Item Image
Item Background Image 1
Item Background Tile Image
Item Background Image 2
Item Foreground Image
Item Floating Image
Item "Has Submenu" Image
Group Background Tile Image
Header Image
Footer Image

There are five tabs in Image Manager:

Project Assets

This tab shows all images that are used (or may be used) by the menu.

You can add images there without necessarily using them in your menu so that you experiment with different images. When you finish your project you can right-click on any image of your Project Assets and click on the "Remove not linked Images" option to remove any images that eventually are not used by the project.

Bullets/Arrows

Bullets are used mainly as Item Image or Item Foreground Image or Item Floating Image.
Arrows are to be used mainly as Item "Has Submenu" Image.

Buttons

Buttons are used mainly as Item Background Image 1, with all other background images being none.
Note that these images are of specific width. If you want "flexible" buttons (i.e. their width to change based on item's content), you should check the **"Sets" tab** instead.

LWBM Buttons

These buttons are created and can be edited by the Likno Web Button Maker (LWBM) application.

If you have installed Likno Web Button Maker, you can simply select an LWBM button and press the "Edit with Likno Web Button Maker" button to launch the related button project there. You can then customize the selected button to your needs:
- change colors
- change textures
- change lighting
- change shape, etc...

In this way you can export different combinations of buttons to your drive and then add them to your Project Assets for use in your menu.
Note that these images are of specific width. If you want "flexible" buttons (i.e. their width to change based on item's content), you should check the **"Sets" tab** instead.

Textures

Textures are used mainly as "Tile" images, i.e. Group Background Tile Image and Item Background Tile Image.

Icons

Icons are used mainly as Item Image or Item Foreground Image or Item Floating Image.

Sets

"Sets" are multiple images applied "as a whole" on certain image properties, such as the item's Background images (Item Background Image 1, Item Background Tile Image, Item Background Image 2) for each of the 3 states ("Normal", "Mouse Over", "Mouse Click").

An image "Set" consists of a "thumbnail" image which defines the set...

*name*_imageset.jpg

...and up to 9 images, with the following **strict** naming:

*name*_imageset_normal_bg1.png
*name*_imageset_normal_bgtile.png
*name*_imageset_normal_bg2.png

*name*_imageset_over_bg1.png
*name*_imageset_over_bgtile.png
*name*_imageset_over_bg2.png

*name*_imageset_click_bg1.png
*name*_imageset_click _bgtile.png
*name*_imageset_click_bg2.png

The ***name_*** part in all filenames identifies the set. The other part should **not** be changed because each word is used by the application to apply the proper image to the respective property.

Example:

Let's assume that a Set called "*set1"* has the following background images for the "Normal" state:

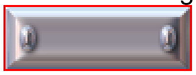
*set1*_imageset_normal_bg1.png

*set1*_imageset_normal_bgtile.png

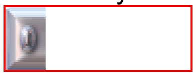
*set1*_imageset_normal_bg2.png

This is how the background of an item shows on "Normal" state, depending on which images of the Set are used (disregard the red border):

When all 3 image files are used (default case):



When only the "*myset*_imageset_normal_bg1.png" image file is used:



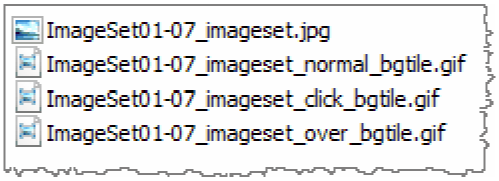
When only the "*myset*_imageset_normal_bg2.png" image file is used:

When only the "*myset*_imageset_normal_bgtile.png" image file is used (tiles in the entire background):

When only the "*myset*_imageset_normal_bg1.png" and the "*myset*_imageset_normal_bg2.png" image files are used:

The following rules apply:

- The first image (***name*_imageset.jpg**) is just the **preview thumbnail** of the Set. It is not applied to any property. It is there so that the user previews the Set and selects it when browsing for Sets. This thumbnail image is strictly in **.jpg** format, the rest of images that are actually applied to properties are in any web format (gif, jpg or png).
- A Set is not required to have all 10 image files listed above. Only the *"name_imageset.jpg"* image is absolutely required, all other images are optional. For example, "ImageSet01-07" uses only the 3 "bgtile" images (animated) without any "bg1" or "bg2" images.



ImageSet01-07_imageset.jpg
ImageSet01-07_imageset_normal_bgtile.gif
ImageSet01-07_imageset_click_bgtile.gif
ImageSet01-07_imageset_over_bgtile.gif

9.5 The Alternative URL

AllWebMenus creates dynamic menus that work in all browsers that support DHTML.

But what about users that have browsers that may not support DHTML? (highly improbable nowadays, except in some handheld devices)

To account for this case, you should specify an **alternative URL** where the user will be **automatically redirected** if he or she uses a browser other than the ones mentioned above. Such page may simply be the web page without the menu, containing additional links to account for the lack of the menu's navigation easiness.

"Where do I specify the Alternative URL?"
The Alternative URL can be specified as an option at the Link Compiled Menu to Web Pages process.
The use of an alternative URL is **not required**.

See also
Link Compiled Menu to Web Pages

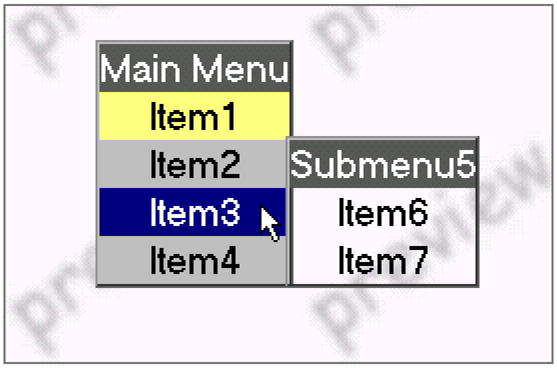
9.6 The different background layers

There are **two** background layers:
1. The **Group background layer** that applies to the area covered by the group.
2. The **Item background layer** that applies to the area covered by a specific item.

A background can be either filled with a color or with an image. If the background layer of the item is set to "none", the background will be transparent and the lower group background will be displayed.

The top-down order of the background layers is as follows:
1. Item Background layer
2. Group Background layer
3. Web page

See the following example:



- Main Menu has a gray background.

The following background rules apply:

Header has a black background
=> black covers gray

Item1 has a yellow background
=> yellow covers gray

Item2 and Item4 have transparent backgrounds
=> Main Menu's gray is displayed

Item3 has a blue background on "Mouse Over" state
=> blue covers gray

- Submenu5 has a transparent background.

The following background rules apply:

Header has a black background
=> black covers transparency

Item6 and Item7 have transparent backgrounds
=> web page is displayed

See also
Example described at <Item Background Image 1> property

9.7 The Paste Values feature

The Paste Values function can be found at the **Edit** -> **Paste Values** menu.

This function is combined with the **Copy** function. When you **Copy** an element (group or item) you can then either **Paste** the entire element inside the menu structure as a new element **or** select a similar element and paste certain properties of the copied element to it by using the **Paste Values** function.

The Paste Values feature is very useful when you want to copy & paste certain style properties from one element to other similar elements.

By using this feature, you can design an element exactly the way you like and then paste all or part of its style properties to other similar elements without affecting their contents (i.e. text, image, etc.). This is very handy when you need to make the same change to the look of many elements.

Note that you can only paste to elements that are similar to the copied element; i.e. if you copied an item you can only paste its values to other items, not to groups.

Older AllWebMenus versions used to have the Pick and Apply functions for a similar functionality. The **Pick** function has now been replaced by **Copy**, while the **Apply** function has now been replaced by **Paste Values.**

9.8 The difference between Main Menu and Submenu

The Main Menu is like a Submenu, consisting of the first group of items that form the initial menu. Therefore, it cannot be called a "submenu" since it does not derive from a parent item.

Apart from the naming problem, there are also some differences that distinguish it from a Submenu:

- The Main Menu cannot be **Cut** or **Deleted**

- Both Main Menu and Submenus can be Horizontal or Vertical but only Main Menu can be an horizontal **MenuBar.**

- Main Menu does not have the following properties that **apply only to Submenus**:
  Appears - When
  Appears - Where
  Appears - Aligned with

9.9 The Element Clipboard

This is where an element (and its children) is stored after performing a **Cut** or a **Copy**. The clipboard understands whether the element is an Item or a Group and will perform a **Paste** according to the following rules:

Element Clipboard -> Selection

- Item -> Item

Pastes the clipboard's Item after the selected Item.

- Item -> Group
  n/a

- Group -> Item
  Pastes the clipboard's Group (and its children items) as a child of the selected Item.

- Group -> Group
  n/a. A Group can only be added to an Item.

See also
Cut, Copy & Paste commands

9.10 The Javascript menu-building Libraries

These are the javascript files containing all the functions that build the menu and drive its behavior. They are named **awmlibX.js** and reside inside the **awmdata** directory which is created upon the Compile Menu process. Only one of them is actually downloaded by the user's browser, the one that is related to that browser.

Their input is the compiled menu file (*menu.js*). Therefore, the *menu.js* represents the definition of the compiled menu whereas the **awmlibX.js** javascript libraries constitute the engine that generates the menu based on its definition.

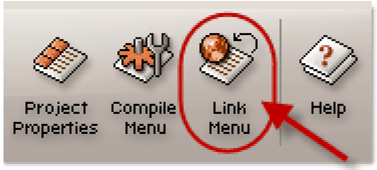
Note for the "CSS Menu" Genre: The *awmlibX.js* files are produced when the default "Javascript Menu" Genre is used. In "CSS Menu" Genre implementations, the ***awmcsslib.js*** menu-building library is produced instead.

9.11 The HTML menu-linking code

This is the AllWebMenus code added to your page's HTML code during the automatic menu-linking process:

```
<!-- DO NOT MOVE! The following AllWebMenus linking code section must always be placed right
AFTER the BODY tag-->
<!-- ******** BEGIN ALLWEBMENUS CODE FOR menu ******** -->
<script type="text/javascript">var MenuLinkedBy="AllWebMenus
[4]",awmMenuName="menu",awmBN="800";awmAltUrl="";</script><script charset="UTF-8"
src="menu.js" type="text/javascript"></script><script
type="text/javascript">awmBuildMenu();</script>
<!-- ******** END ALLWEBMENUS CODE FOR menu ******** -->
```

Depending on the menu name and path you use, your code may be different. Click on the "Link Menu" icon on the toolbar to see your own one.



Notes:

- The menu-linking code does **not** determine where the menu appears on page (this is specified through the <Positioning> property). Therefore, do not think that if you move it from right AFTER the <BODY> tag that will have any effect on menu positioning.

- The menu-linking code should be a **direct child** of the <BODY> tag. If this is not possible by your CMS or website builder, you need to inform AllWebMenus by selecting the related option at the "Menu-Linking Code in Tags" property, otherwise the menu may not appear correctly.

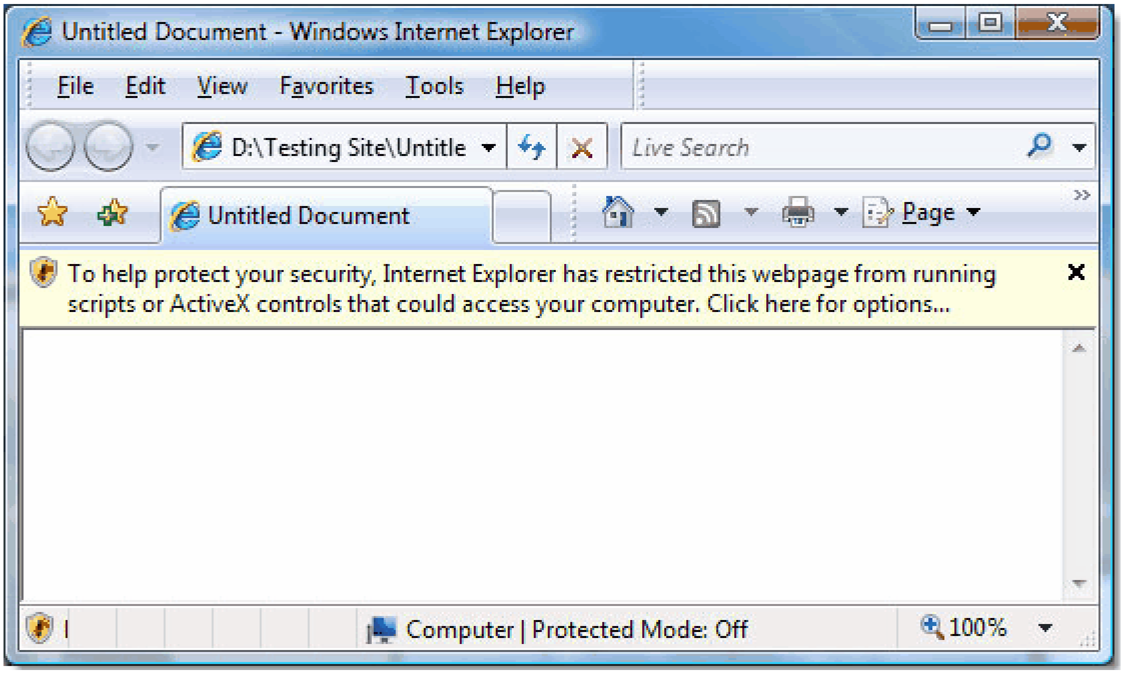
Note for users of the "WordPress Menu Add-in", "Joomla Menu Add-in", etc.:
The menu-linking code is **different on these environments!**

- Find the menu-linking code and instructions for **WordPress Menus** here:
http://www.likno.com/addins/wordpress-menu.html

9.12 The Internet Explorer Information Bar Warning

When you preview a **local** web page in Internet Explorer that contains **javascript** (like our menus do), you will get the following message:

[To help protect your security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your computer. Click here for options...]



To help protect your security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your computer. Click here for options...

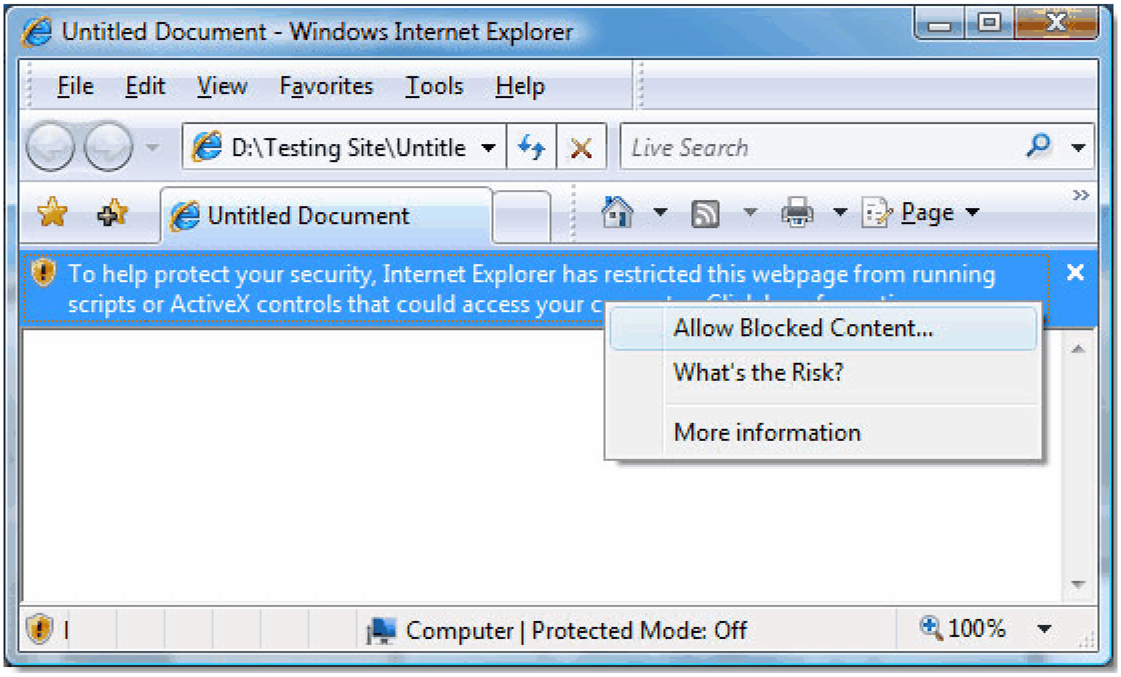
This is **not a problem**, as the "security" warning **never appears ONLINE on your actual pages** but only locally, for the pages that are located at your PC.

Furthermore, this is not related to AllWebMenus but to **all javascripts** running on your pages **locally**.

This default behavior of Internet Explorer only happens when viewing pages saved to the **local** computer and only in Internet Explorer.

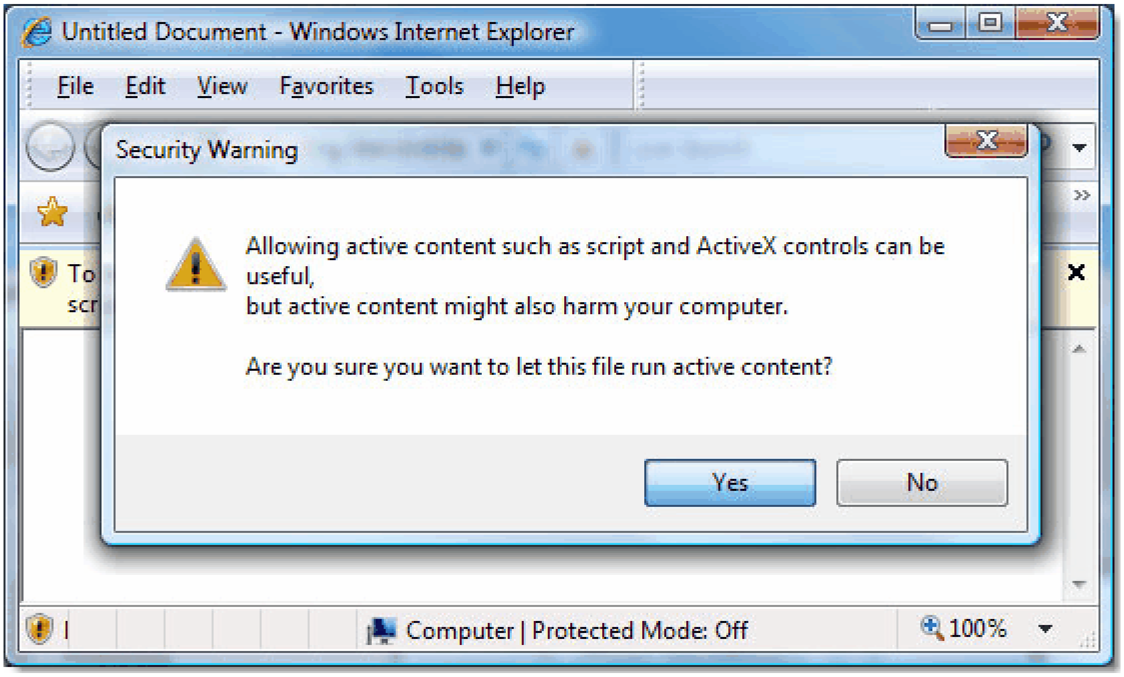
This does not happen when pages have been uploaded to a web server and are then viewed online.

To overpass this Internet Explorer warning locally, you can click on the message bar and select **"Allow Blocked Content"**:



To help protect your security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your c

Then, a warning dialog will appear and you can just press **"Yes"** for the web page to properly load all the menu related files:


Untitled Document - Windows Internet Explorer
File
Edit
View
Favorites
Tools
Help
Security Warning
Yes
No
Computer | Protected Mode: Off
100%

Allowing active content such as script and ActiveX controls can be useful,
but active content might also harm your computer.

Are you sure you want to let this file run active content?

(do not worry about the warning, our scripts are 100% safe)

You can read more about the IE Information bar at:
http://support.microsoft.com/kb/843017 (Windows XP)

http://windowshelp.microsoft.com/Windows/en-US/Help/91b1038d-3f53-43a3-8bd8-9551e7c0b7c61033.mspx (Windows Vista)

FAQ and Tips

Part X

10 FAQ and Tips

10.1 Troubleshooting

This is a collection of quick tips to avoid trouble.

Note: An Online Knowledge Base is also available with more information and up-to-date content. Please check this knowledge base also.

Menu appears in the top left corner when the page loads, and then "jumps" to its specified location.
Change the <*Position*> property to “Element” and recompile.

Compile Menu - Link Menu process:

- It is critical that you do not forget to upload the **awmdata** directory along with the ***menu*.js** file after you compile your menu.

- You can compile the menu anywhere you like **within** your site's directory structure. This decision should not be affected by the location of the pages you want to link the menu to.
  For example, if you want to link the menu into a Dreamweaver template, you are **not** required to compile the menu into the Templates directory. On the contrary, we do not advise you to do so as you will have to upload the Templates directory on the server too.
  In general, simply compile your menu into your root directory or into a manually-created subdirectory and you will be able to link it to any page, shared border, template or library of that web site directory structure you want.

- If you use **FrontPage**, the menu should not be compiled inside the *_borders* folder or any other special folders used by the authoring tool to manipulate common objects. The obvious reason is that these folders are not uploaded on the server when publishing the site. Instead, compile the menu in the root or in any other manually-created subdirectory.

Restore an old menu.awm project file, originally made with AllWebMenus v1+
When you convert the old project file to version 4, the old version of AllWebMenus won't recognize it. However a “.bak” file is immediately being generated in the same directory, after you select yes, to the project conversion dialog box. This is a backup copy of the old file.

Font selection:
For cross-platform compatibility, it is advisable to use the generic font families instead of any system-specific font. For example, if you use "Arial" and a specific platform does not have the Arial font, then the menu will use the browser's **default** font. But if you use "Sans Serif" instead of "Arial", the browser will use the appropriate Sans Serif font (in most cases it will be "Arial", but in platforms that do not have Arial it will be a **similar** font; not the default font)

Filenames:

- **Do not use filenames that contain spaces** (always use "my_image.gif" or "myimage.gif" instead of "my image.gif") **or special characters**, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

- **Spaces in the compiled menu name are not allowed** as they are not parsed properly by some browsers. As an example, use "main_menu.js" or "mainmenu.js" instead of "main menu.js".

Note on <*Javascript Command*> syntax:
You may use only quotes (") or only apostrophes (') inside the text of the property but **not both**! Using both quotes and apostrophes will cause a javascript error.

Note on <*Link*> syntax:
Always use **slashes** instead of backslashes
(example: ../users/~user10/user10.html)

10.2 Correct Placement of the menu-linking HTML code

It is important that the AllWebMenus menu-linking code is placed right after the <BODY> tag and not somewhere else. This allows for the floating menus to appear properly on the page without being dismantled by other page elements (such as tables etc.), mostly in Gecko-based browsers.

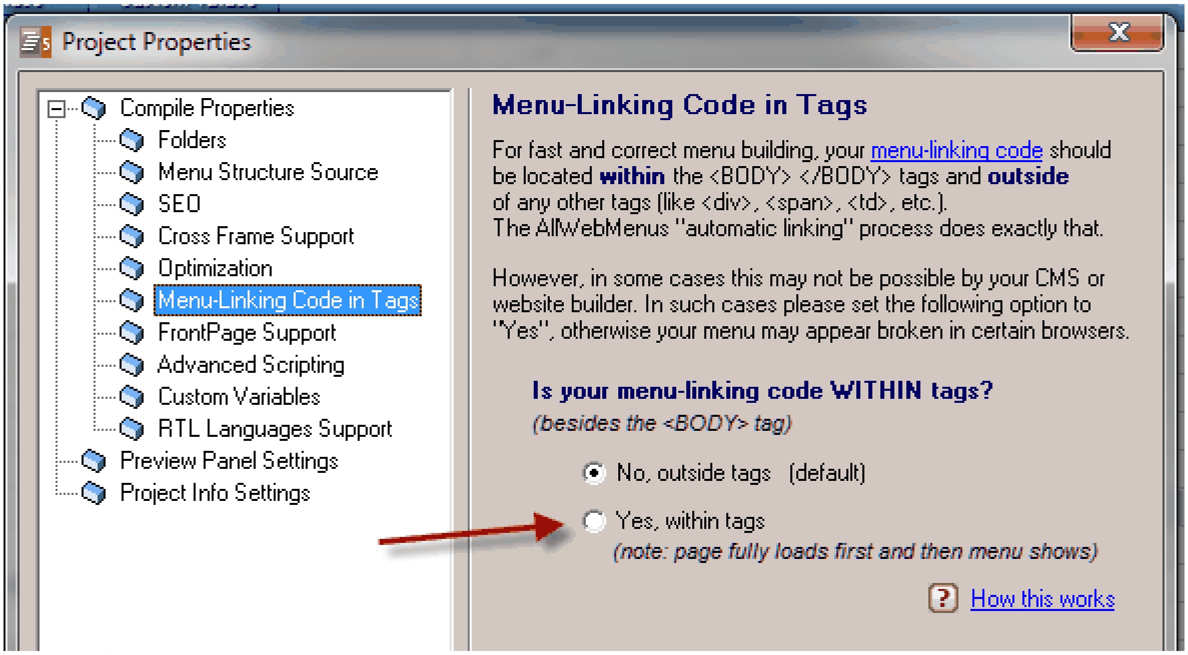
How it works:
When inserting a menu into a web page, something similar to the following code is added right after the <BODY> tag:

```
<!-- DO NOT MOVE! The following AllWebMenus linking code section must always be placed right
AFTER the BODY tag-->
<!-- ******** BEGIN ALLWEBMENUS CODE FOR menu ******** -->
<script type="text/javascript">var MenuLinkedBy="AllWebMenus
[4]",awmMenuName="menu",awmBN="628";awmAltUrl="";</script>
<script charset="UTF-8" src="menu.js" type="text/javascript"></script><script
type="text/javascript">awmBuildMenu();</script>
<!-- ******** END ALLWEBMENUS CODE FOR menu ******** -->
```

This menu-linking code should always remain there no matter where you want the menu to appear. The menu's appearing position is **not** determined by where you place the AllWebMenus menu-linking code but by specifying and implementing the positioning methods found at the <Positioning> property of its Main Menu group.

Therefore, please **do not move** the menu-linking section from the start of the page.
If you do so, you may have problems with some browsers!

If you cannot do otherwise, make sure that you check this option in "Project Properties":



Menu-Linking Code in Tags

For fast and correct menu building, your menu-linking code should be located **within** the <BODY> </BODY> tags and **outside** of any other tags (like <div>, <span>, <td>, etc.). The AllWebMenus "automatic linking" process does exactly that.

However, in some cases this may not be possible by your CMS or website builder. In such cases please set the following option to "Yes", otherwise your menu may appear broken in certain browsers.

Is your menu-linking code WITHIN tags?
(besides the <BODY> tag)

● No, outside tags (default)

○ Yes, within tags
(note: page fully loads first and then menu shows)

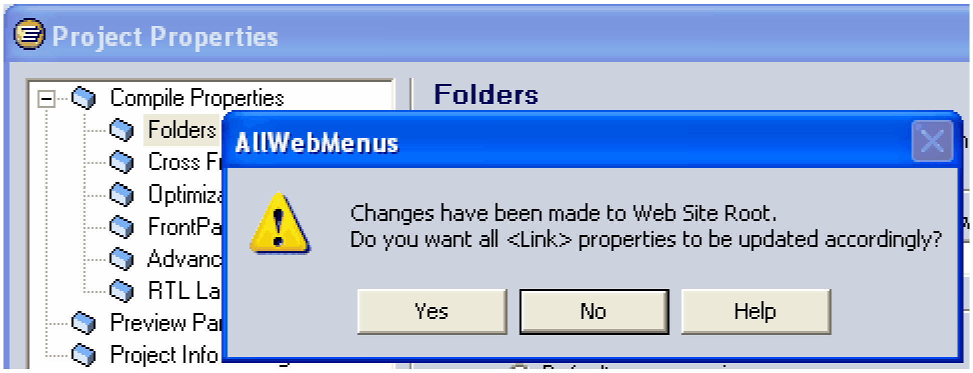
(use this option only if you have to)

What about Includes, Libraries, Templates, etc.?
The same rule applies to includes, libraries and templates. If you want to have a menu into a library or an include then you should make sure that:
1. the menu is always at the very beginning of the library or include **and**
2. the library or include is the first element of the page, right after the <body> tag.
If both conditions are satisfied, the resulting page will have the AllWebMenus menu-linking code at the very beginning of the page and no problems will occur in Gecko-based browsers.

What about FrontPage Shared Borders?
A similar logic applies to the use of FrontPage Shared Borders.
This topic is best described in the "Compile Properties, FrontPage Support" section.

10.3 Updating all <Item Link> properties when changing the Web Site Root folder

When updating the **Web Site Root** folder, the following message may appear:



Changes have been made to Web Site Root.
Do you want all <Link> properties to be updated accordingly?

What is this?

As described in the <Item Link> help topic, your menu items may link to pages using **relative paths**. These links are relative to the specified *Web Site Root* folder.

But what happens if you change the *Web Site Root* to a different folder? In that case your links will not be able to open the pages because the path to the root folder is changed.

For example, if your *Web Site Root* folder is **c:\mysite**, the link to the page **c:\mysite\products.htm** would be **products.htm**.
If you then change the *Web Site Root* folder to **c:\mysite\subfolder**, the link to the page **c:\mysite\products.htm** should be updated to **../products.htm**.

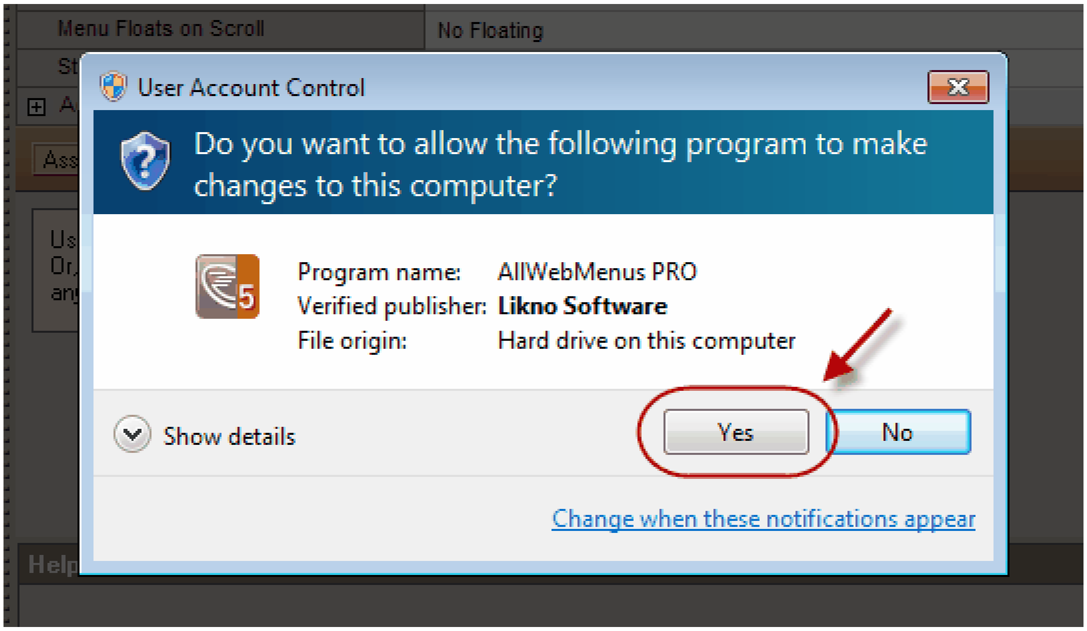
This is why you may want to press the [Yes] button in the above message, so that AllWebMenus automatically updates all <Link> properties to the correct path to the *Web Site Root* folder.

10.4 "Full Admin rights" and the activation process

In order to activate AllWebMenus you need to be an "administrator" on that PC. Otherwise, the activation process fails.

There are some cases (Windows Vista or 7 mostly) where you might think that you are administrator but you miss certain rights that are required by the activation process (you may be a "protected administrator' instead).

To get "full" administrator rights for the activation process,choose **"Yes"** on the following popup window:



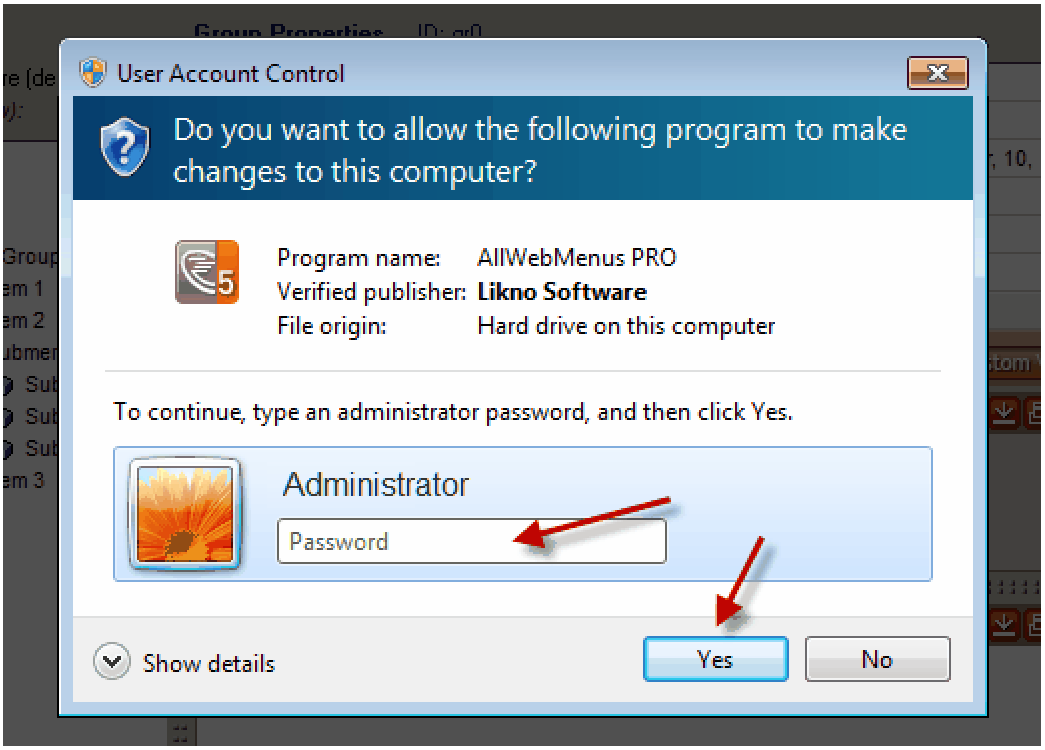
User Account Control

Do you want to allow the following program to make changes to this computer?

Program name: AllWebMenus PRO
Verified publisher: **Likno Software**
File origin: Hard drive on this computer

(this window appears only for the activation process, you will not see that again after you activate)

If an administrator password is requested and you do not have this, then please contact your administrator to activate AllWebMenus through the administrator account.



User Account Control

Do you want to allow the following program to make changes to this computer?

Program name: AllWebMenus PRO
Verified publisher: **Likno Software**
File origin: Hard drive on this computer

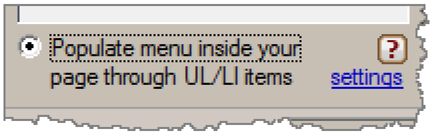
To continue, type an administrator password, and then click Yes.

Administrator

10.5 Mouse Click state ignored

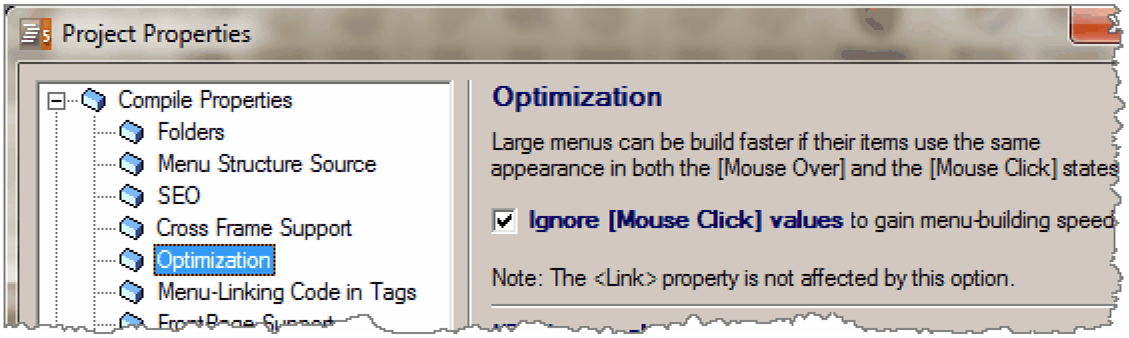
The "Mouse Click" state is not applicable and all its values are ignored when **any** one of the following apply:

1. The **"Populate menu inside your page through UL/LI items"** option is selected in Menu Editor.



● Populate menu inside your page through UL/LI items

2. The **"Ignore [Mouse Click] values"** is checked in the **Optimization** property.



Optimization

Large menus can be build faster if their items use the same appearance in both the [Mouse Over] and the [Mouse Click] states

☑ **Ignore [Mouse Click] values** to gain menu-building speed

Note: The <Link> property is not affected by this option.

(in this case the Item Link property is the only "Mouse Click" value that is not ignored)

So what happens in these cases when the user clicks on the menu items? Their "Mouse Over" values are used instead.

Frequently Asked Questions

Part XI

11 Frequently Asked Questions

Note: An Online Knowledge Base is also available with more information and up-to-date content. Please check this knowledge base also.

Do I have to place a link from my website to www.likno.com in order for my menu to work?

No, this is not required. AllWebMenus is CLEAN of any hidden links! This "strategy" is applied by most of our competitors for boosting their own page rank, which consequently makes yours drop (or can even get your website penalized)!

We use clean javascript, optimized for each browser separately, which does not affect negatively your website ranking.

My menu works fine when viewed locally but does not appear when viewed on the server. What have I done wrong?

You have forgotten to upload the associated **awmdata** directory (and its content) along with the compiled menu or you have uploaded them in the wrong directory.
Click here to read more about this procedure

How can I optimize my menu design to get the best possible menu-building speed?

Click here to read important notes on the Performance Optimization

Can I use HTML tags inside the Items and Titles?

Yes, you can use any HTML tag you like inside the *<Item Text>, <Header Text> or <Footer Text> properties*. This is very useful when you want to break the text into several lines. As the menu in our web site shows, the "Frequently Asked Questions" item is broken into three lines by assigning the value "**Frequently Asked Questions**" to the Item Name property.

Can a menu item link to a paragraph within a web page?

Yes, just enter **http://www.my-domain.com/info.html#anchor** in the <*Link*> property of the items and the menu can be used to navigate inside the **info.html** web page.

Can I associate one of the menu Items (eg. CONTACT US) with my email program such as Microsoft Outlook?

You can add "mailto:login@domain.com" to the URL property of your Menu Item:

I have two different menus for different html pages. How do I properly use the different compiled menus and their associated *awmdata* directories?

Let's assume the following file structure:
c:\mysite\pageA.html
c:\mysite\pageB.html
c:\mysite\pageC.html

c:\mysite\pageD.html
(the ***c:\mysite*** directory is the root of the site)

and the following designed menus:
menu1.awm (to be used by pageA and pageB)
menu2.awm (to be used by pageC and pageD)

1. We compile **menu1.awm** into the ***menu1.js*** file.
This action creates the following elements into the c:\mysite\ folder:
-the ***menu1.js*** file
-the ***awmdata*** directory which contains the javascript libraries and all images used by menu1 (if any)

2. We compile **menu2.awm** into the ***menu2.js*** file.
This action creates the following elements into the c:\mysite\ folder:
-the ***menu2.js*** file
-the ***awmdata*** directory which contains the javascript libraries and all images used by menu2 (if any)
(be careful not to use same image names in both menus, as the one image will be copied above the other one inside the awmdata folder).

3. We link the menu1.js file to pageA and pageB

4. We link the menu2.js file to pageC and pageD

5. We upload the following elements to the root of our web site:
-the ***menu1.js*** file
-the ***menu2.js*** file
-the ***awmdata*** directory
-the ***pageA.html*** file
-the ***pageB.html*** file
-the ***pageC.html*** file
-the ***pageD.html*** file

6. This completes the process for all menus and pages.

Note: The **awmmenupath.gif** file is obsolete if your menu is linked with **version 4.2** or above.

Important note on the "Link" property for .awm files created with version 1.2 or earlier:

The older versions created a menu entity (i.e. the awmData subdirectory) in **every** directory where the pages were located. Therefore, there was no actual difference between "document-relative" links and "menu-relative" links in the URL property (eg: "../info.html")

The new version creates a **single compiled menu** for all pages, regardless of where these pages are located inside the site's directory structure. Therefore, all relative links are now related to the directory where the compiled menu is located. In that manner, the old "../info.html" value may not link to the proper page now and it may have to be changed to "info.html" (or whatever needed).

Obviously, you will **not need** to do any changes if you have been using **absolute paths** inside the URL property (eg: http://www.likno.com).

See more info on the <Link> property here

Index

Likno software

http://www.likno.com