

User's Guide

Likno Web Modal Windows Builder

The Complete Tool for Creating Powerful Web Modal Windows

by Likno

Likno Web Modal Windows Builder is a powerful jQuery interface (GUI) for creating stylish and feature-rich javascript popup windows/dialogs for your web pages, with minimal effort and coding.

The present manual will assist you in taking the most out of this software.

The Likno Software team

Likno Web Modal Windows Builder User's Guide

© 2000-2014 Likno Software, all rights reserved

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

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.

The javascript code used by this software application is property of Likno Software. You may use this code only as part of this software for creating web projects. Any unauthorized use, transmission, transformation, facsimile, or copying of any means (electronic or not) is strictly prohibited.

Removal of the present copyright notice is prohibited.

Printed: 10-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 Likno Web Modal Windows Builder 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!

Table of Contents

Part I Introduction	2
1 What is Likno Web Modal Windows Builder	2
2 What's new	3
3 Terminology	3
4 About Likno Software	4
Part II A Web Modal Window in 5 easy steps	6
1 Step 1: Select the content and properties for each Modal Window	6
2 Step 2: Select the Project Style of your Modal Windows	6
3 Step 3: Compile and Link the Project	6
4 Step 4: Upload the files to your server	7
5 Step 5: Save your work	7
Part III User Interface	9
1 Main window layout	9
2 Keyboard shortcuts	9
Part IV Commands	12
1 Command: Project	12
2 Tools	12
Command: Compile Project	12
Command: Link Compiled Project to Web Pages	13
Command: Project Properties	14
Compile Properties.....	14
Preview Panel Settings.....	16
Under the Hood.....	17
3 Command: Help	18
Part V Properties	20
1 Properties Overview	20
2 "Project Style" Properties	20
Modal Window Opens - How	20
Modal Window Opens - When	21
Modal Window Closes - How	21
Modal Window Closes - When	22
Modal Window Position	23
Modal Window Width/Height	25
Modal Window CSS	26
Modal Window Z-Index	27
Modal Window Rounded Corners	27

Modal Window Shadow	30
Overlay Color	31
Overlay Opacity	31
Header	31
Body CSS	32
Footer	33
AJAX "Loading" Info	34
Multi-Sheet properties	34
3 "Project Windows" Properties	37
Make certain links/forms in a page open in this window	37
Show Content as specified inside this project	47
Sheets	48
Content Source options for a single-sheet window	49
Embedded	49
Page Element	50
Image	50
AJAX	51
External	51
Content Source options for a multi-sheet window	52
Embedded	52
Page elements	54
HTML File	55
Opens - When.....	56
Opens on Page Load	65

Part VI How Do I... 71

1 Find my INSTALLATION ID	71
2 Purchase/Activate	71
3 Use the application with my favorite Web Authoring tool (Dreamweaver, Expression Web, etc.)	71
4 Use an already existing element of my page as a Modal Window	72
5 Add effects when the Modal Window opens or closes	72
6 Change the position where the Modal Windows appear	72
7 Define a Header for my Modal Windows	72
8 Define a Footer for my Modal Windows	72
9 Load with AJAX the content of a Modal Window	73
10 Load an external site as the content of a Modal Window	73
11 Make my modal window appear over a Flash element	73
12 Use multiple Modal Window projects in the same page	74
13 Link a Modal Window to an item of an AllWebMenus menu in my page?	74
14 Display a video/movie in a Modal Window?	74
15 Properly use paths in the "Website Folder" option	75
16 Use the "background-image" value in CSS properties	76

Part VII What is... 80

1 The HTML Linking Code	80
2 The Header	80

3	The Footer	80
4	The Internet Explorer Information Bar Warning	81
5	The "Colors detected in this CSS" feature	83
6	The "Relative paths detected in this HTML" feature	83
Part VIII FAQ and Tips		86
1	Troubleshooting	86
2	"Full Admin rights" and the activation process	86
3	Modal Windows opened from Frames or IFrames	88
Part IX API		90
	Index	95

Introduction

Part



1 Introduction

1.1 What is Likno Web Modal Windows Builder

[Create Web Modal Windows in 5 easy steps!](#)

Likno Web Modal Windows Builder is an application that serves as an interface to the popular **jQuery** open source library, allowing you to create various "Modal Windows" for web pages with minimal effort and coding.

It creates **cross-browser** jQuery modal windows that work alike in all browsers.

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.

Through our builder you can easily create web modal windows with many options, like showing internal HTML content (located in your actual page), content added easily through the interface (embedded), external or internal (AJAX) pages, and more.

The produced modal windows are fully cross-browser, i.e. they work alike in all browsers supporting DHTML (like Firefox, Internet Explorer, Google Chrome, Opera, Mozilla, Konqueror, Safari and more).

Likno Web Modal Windows Builder will save you a significant amount of time and effort required to fully understand DHTML and Javascript code capable to support all browsers for creating powerful web modal windows!

Major Features Include:

- * Rich and powerful interface!
- * Create cross-browser jQuery Modal Windows to display information when visitors click on page elements (or even on page load!).
- * Use simple text, HTML code, images or whole pages as the content of your Modal Windows.
- * Customize the style of your Modal Windows through CSS.
- * Use multiple Modal Windows on the same page.
- * Add dynamic content (database-driven or programmable) to your Modal Windows through AJAX techniques.
- * Use page elements to define the content of your Modal Windows easily from within the page.
- * Open and close your Modal Windows using a variety of opening/closing effects.
- * Add a Header or Footer to your Modal Windows for extra information.
- * Create projects fast by using the ready-made examples already installed in the application.
- * Create single-sheet Modal Windows (all content on a single sheet) or multi-sheet Modal Windows (multiple content on separate sheets, for an Image Gallery, Product Presentation, etc.).
- * Use a variety of Navigation Bar styles on multi-sheet Modal Windows.
- * Open Modal Windows through a variety of triggering options: TARGET, ID, CLASS, onclick event, AllWebMenus item.
- * Set a Modal Window to open automatically on page load, instead of clicking on a page element.
- * Use complete help for all properties and procedures. Press F1 while you are on a property to immediately receive Help for that property.
- * Live Examples available online.
- * Powered by the popular jQuery open source library.

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

1.2 What's new

Please check the new features, enhancements and fixes of the application by visiting the following page online:

<http://www.likno.com/jquery-modal-windows/whatsnew.php>

1.3 Terminology

Modal Window

"A modal window is a child window that requires the user to interact with it before they can return to operating the parent application, thus preventing the workflow on the application main window" (source: http://en.wikipedia.org/wiki/Modal_window).

A modal window is a part of a **Modal Window project**; all windows of this project have a similar functionality and appearance that is defined in the "Style Editor" tab, as "Project Style".

Modal Window Project

All modal windows contained within a Project share the same "Project Style" properties and are compiled in a single project file. A project can contain several modal windows to be used on a single web page or different web pages of the same website.

If you want to have modal windows in separate projects read more information [here](#).

Overlay

It is the layer that appears on top of the page and behind the modal window that prevents the interaction with the rest of the page, bringing the focus only to the modal window.

Content

The Content of the modal window is the information that displays. This content can be from a simple text, to rich HTML content, and even a whole new page derived either as an AJAX call or as an external website.

More information [here](#).

Header

Adds text or HTML code as a header to all the modal windows of the project.

More information [here](#).

Footer

Adds text or HTML code as a footer to all the modal windows of the project.

More information [here](#).

1.4 About Likno Software

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.

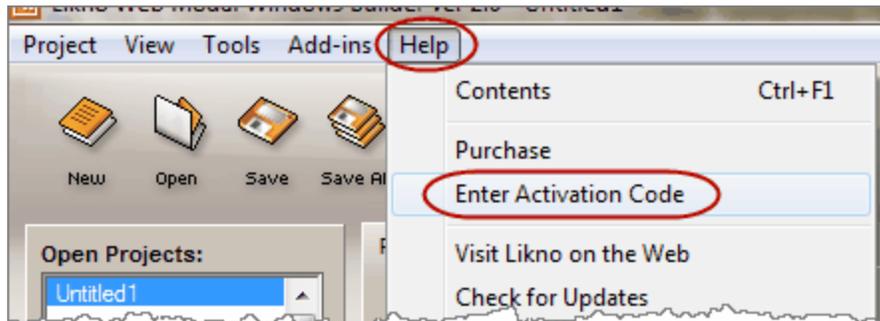
For detailed **technical support** you can access the following URL:

http://www.likno.com/support_page.html

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

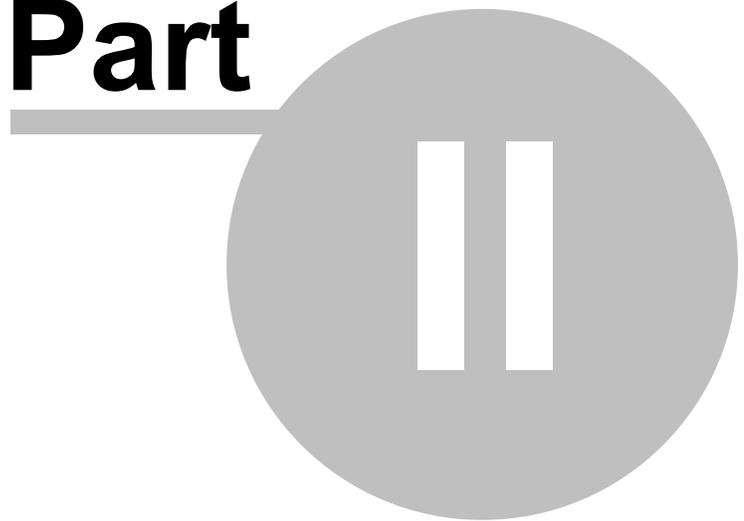
Our web site is located at <http://www.likno.com>

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:



A Web Modal Window in 5 easy steps

Part



2 A Web Modal Window in 5 easy steps

2.1 Step 1: Select the content and properties for each Modal Window

Step 1

In the **<Project Windows>** tab you can find all the editable properties for each modal window individually.

All properties are described and explained in the "[Properties](#)" help section.

Also read: [Detailed integration info with your website \(PDF\)](#)
[FrontPage-specific integration info \(PDF\)](#)

2.2 Step 2: Select the Project Style of your Modal Windows

Step 2

In the **<Style Editor>** tab, choose the properties that will apply to all the modal windows of the project.

All settings are described and explained in the "[Properties](#)" help section.

Also read: [Detailed integration info with your website \(PDF\)](#)
[FrontPage-specific integration info \(PDF\)](#)

2.3 Step 3: Compile and Link the Project

Step 3

Compile:

Click on "Compile" button and select the location where you want to save your compiled JavaScript files.

Link:

Once you have compiled your project, use the "Link" button to retrieve the necessary code that must be added in your pages in order to link the compiled files to them.

Make sure that the correct path to the source javascript file has been used, otherwise the modal windows will not appear.

Also read: [Detailed integration info with your website \(PDF\)](#)
[FrontPage-specific integration info \(PDF\)](#)

2.4 Step 4: Upload the files to your server

Step 4

Once you have compiled and linked your compiled javascript project to your pages, you will need to upload the files to your web server. The files created by Likno Web Modal Windows Builder during the compilation phase are:

```
./likno-scripts/                (folder)
./likno-scripts/likno-modal-project.js  *
./likno-scripts/likno-modal-lib.js
./likno-scripts/jquery.js
./likno-scripts/likno-modal-project/    * (folder)
(all image files used by the modal windows)
```

where `./` is the "Site_Root" folder (also specified in Project Properties).

(* the highlighted names may be different if you changed the "Compiled Javascript Project" name in Project Properties)

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 set while linking your compiled project locally to point to the right structure on your web server.

Also read: [Detailed integration info with your website \(PDF\)](#)
[FrontPage-specific integration info \(PDF\)](#)

2.5 Step 5: Save your work

Step 5

Your Web Modal Windows can be saved in an `.wmw` project file so that you can use/edit it in the future. Just use *Project->Save* or *Project->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 *likno-modal.js* file to your web server. The *likno-modal-lib.js* file needs to be re-uploaded only if you re-compile your project with a newer Likno Web Modal Windows Builder version (since it may use an updated modal window-building library).

Also read: [Detailed integration info with your website \(PDF\)](#)
[FrontPage-specific integration info \(PDF\)](#)

User Interface

Part



3 User Interface

3.1 Main window layout

Likno Web Modal Windows Builder uses several tabs:

Project Windows tab

Here you can add as many modal windows as you need in your project and specify their content.

Style Editor tab

Here you can modify the style properties that apply to the modal windows of the loaded project.

Preview tab

Here you can view all the modal windows that you have defined in the Project Windows tab - the default that will appear is the modal window selected in the "Modal Windows in Project" list. Close this Modal Window and click on any link to view the rest of the Modal Windows.

Project Properties tab

Here you can specify the compiled files and folders of the project, as well as the domain which the project is compiled for.

3.2 Keyboard shortcuts

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

- **F1**
Provides context sensitive help for each property
- **F2**
Opens the Project Windows tab
- **F3**
Opens the Style Editor tab
- **F5**
Refreshes and opens the Preview tab
- **F6**
Opens the Project Properties tab
- **Ctrl + F12**
Opens the Compile dialog

- **F8**
Compiles the project without any dialogs
- **Shift + Ctrl + F12**
Opens the "Link Compiled Project to Web Pages" dialog
- **Ctrl + N**
Creates a new project
- **Ctrl + O**
Opens a .wmw project
- **Ctrl + S**
Saves the selected project

Commands

Part



4 Commands

4.1 Command: Project

New

Creates a new **Likno Web Modal Windows** project.

Open

Opens a Likno Web Modal Windows project.

The Likno Web Modal Windows files are found under the **.wmw** extension.

Save

Saves the selected project into a ***.wmw** file.

Save As

Saves the current project with a different filename.

Save All Projects

Saves all loaded projects, but not the examples which are read-only.

Close Selected Project

Closes the selected project.

Close All Projects

Closes all loaded projects, but not the examples which are always available.

Export Project

Exports the loaded project as a ZIP file in order to move it to other PCs without having problems with the file paths of the linked images (the "assets" folder).

Exit

Closes the **Likno Web Modal Windows Builder** application.

4.2 Tools

4.2.1 Command: Compile Project

Compiles the **.wmw** project into a **.js** file that has to be linked to your web pages.

To use modal windows in your web pages you must take the following four steps:

Step 1. **Compile** the project (note: spaces are not allowed in filename)

Step 2. **Link** the compiled project to your web pages

Step 3. **Upload** the necessary files and folders (see a description of these elements below)

Step 4. **Upload** the web pages that contain the project link

(note: steps 2 & 4 are not necessary if you have already linked the project to your web pages and you only want to update your modal windows)

How it works:

When compiling a project as *likno-modal-project.js*, the following elements are created inside the same directory:

1. The core **likno-modal-project.js** javascript file.
2. The "Compiled Libraries" and "Compiled Images" folders as defined in [Project Properties](#).

It is critical that **you do not forget to upload the likno-scripts directory**, where these files are compiled.

Also read: [Detailed integration info with your website \(PDF\)](#)
[FrontPage-specific integration info \(PDF\)](#)

See also

[Link Compiled Project to Web Page\(s\)](#)

4.2.2 Command: Link Compiled Project to Web Pages

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

At this stage, the application inserts the necessary link into the selected web pages. No DHTML or Javascript knowledge is required since the whole process is conducted by the application. The 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 project file** (eg: likno-modal-project.js)
2. The **web page(s)** where the link will be added.

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

1. the compiled project file (eg: *likno-modal-project.js*)
2. the "*likno-scripts*" folder
3. the "*likno-scripts/likno-modal-project*" folder if you use images. Your folder may be different depending on the name you choose to use for the compiled project file.

...at the **same** directory structure where they reside locally.

See the [Compile Project](#) command for more info on this issue.

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

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 project 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 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 project 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 project into an **include** so that it can be used by all other pages. Just make sure that this include contains only the project and that it is inserted immediately before the </body> tag of the pages.

Also read: [Detailed integration info with your website \(PDF\)](#)
[FrontPage-specific integration info \(PDF\)](#)

See also

[Compile Project](#)

4.2.3 Command: Project Properties

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

[Compile Properties](#)

[Preview Panel Settings](#)

4.2.3.1 Compile Properties

Allows you to specify the following compile properties for your loaded 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 as it is used in the compile process to export the project files there.

The **Compiled Javascript Project** is the javascript file generated when you compile your project. It defaults to *likno-modal.js* but you may as well change it to something else. Note that you can use multiple modal projects in the same website by having different projects compiled using different "Compiled Javascript Project" names (do not forget to **link** each project individually).

Once you setup your "Site_Root folder", you are able to edit the "**Compiled subfolders**":

The **Compiled Libraries subfolder** (default: \likno-scripts) which contains the following:

likno-modal-lib.js The **required** Javascript modal-building Libraries. These are the javascript files containing all the functions that build the modal windows and drive their behavior.

likno-modal-project.js The compiled project file (actual name depends on your settings)

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

The **Compiled Images subfolder** (default: \likno-scripts\likno-modal-project) which contains any image files used by your modal windows. If not uploaded, the modal windows will show no images. The missing image sign ("**x**") may also appear in place of those images.

Custom folders may be required when two or more different modal projects are used in the same website and images from both projects use the same names. This is the usual situation that may require you to have a different subfolder for the produced images.

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

The **Domain (website) where the compiled project will be used** is very important as it affects whether your modal windows are available to your online visitors or just you (for testing purposes).

You can always view (test) your modal windows by using a **localhost** or **numeric IP** on the browser but not when using a "**letter-based**" domain (eg:www.fifa.com), **unless** you have **purchased a domains license** and allocated a domain to that license.

For example, "**fifa.com**" uses the "**83.231.250.129**" numeric IP.
(find the numeric IP of any domain name here: <http://www.hcidata.info/host2ip.cgi>)

FIFA developers could then test their modal windows using any of the following:

c:/portfolio/fifa_site/page.html (local hard drive)
http://localhost/page.html (localhost)
http://83.231.250.129/page.html (their numeric IP)

But their actual visitors could **not view** the modal windows when using the respective "letter-based" domain:

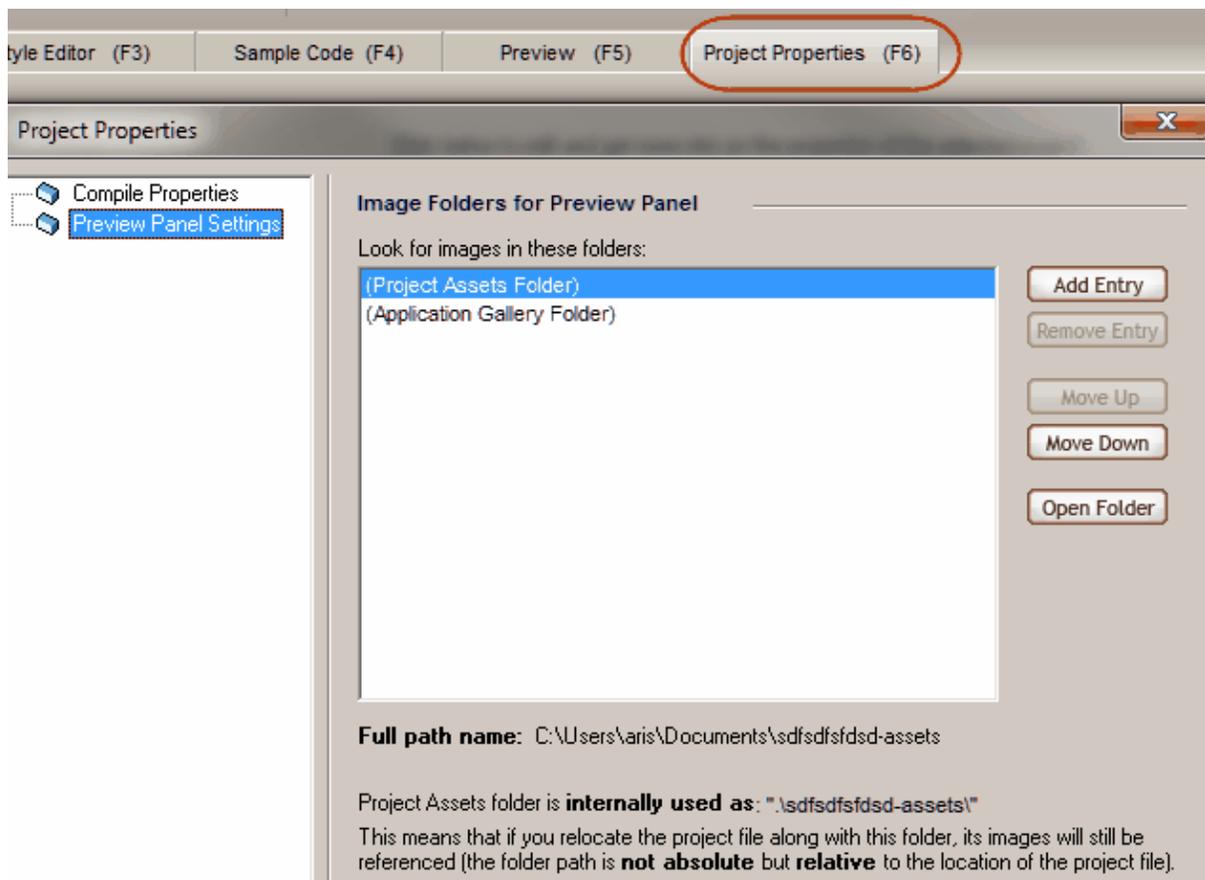
http://www.fifa.com/page.html (domain)

So, what FIFA could do is purchase the "3-Domain" license, **allocate "fifa.com"**, get an Activation Code and choose to compile for the "fifa.com" domain. Or simpler, purchase the **"Unlimited Domains"** license, get an Activation Code and just compile for ANY domain. Any of these 2 options would make the windows available for the "fifa.com" online visitors.

Note on "subdomains": Do not allocate subdomains, just allocate their major domain, as **10 of its first-level subdomains** will be supported by default (examples: "subdomain.fifa.com", "subdomain2.fifa.com", but **not "sub.subdomain.fifa.com"**). For more than 10 subdomains (or second-level subdomains) on the same domain, you need to purchase the "Unlimited Domains" License.

4.2.3.2 Preview Panel Settings

Allows you to specify folders so that the Preview Panel can show images (if used) that are **relative to the page or site root**.



While these images will show online (on the actual site), the **local Preview Panel** cannot show them, unless you specify local folders to browse from.

For previewing purposes, the Preview Panel locates graphics on the disk by searching for files in the directories in the **"Image Folders for Preview Panel"** list.

By default the "Project Assets Folder" and the "Application Gallery Folder" are included in the Image Folders list so that any images in those folders will be found automatically. You cannot remove them from the list.

Please note that you must be careful to **avoid duplicate graphics file names in different folders** (the standard folders or yours). If this occurs, the 1st match will "win" depending on how the order of folders is (use the "move up/down" buttons to determine the priority of folders).

4.2.3.3 Under the Hood

Allows you to specify the behavior for special cases.

If the browser window shows a scrollbar you can choose to hide this scrollbar when the Modal Window appears, or not.

4.3 Command: Help

[Contents](#)

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

[Enter Activation Code](#)

This is where you can enter an Activation Code. You will receive this Activation Code through your Likno User Area account which is created after your purchase of a license for the application.

[Visit Likno on the Web](#)

Visit our web site to contact us, get the latest information on new versions and products.

[Check for Updates](#)

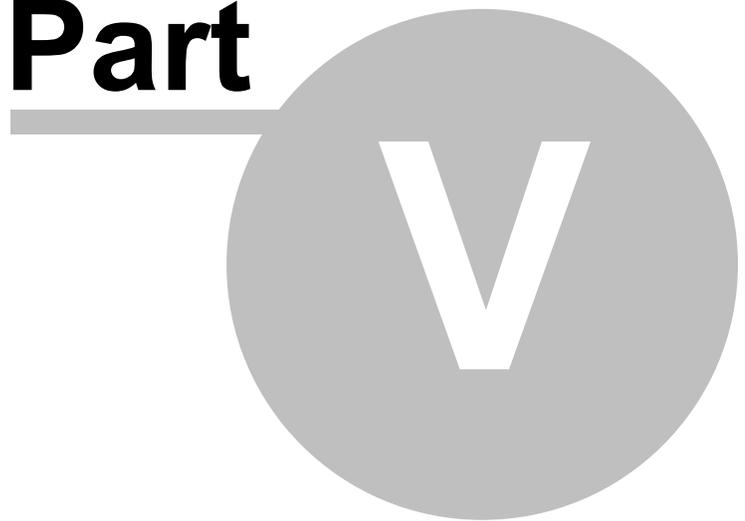
Checks if newer versions have been released since the last time you installed the application. It is recommended that you check for updates frequently as we regularly release improved cross-browser support for your implementations.

[About](#)

Displays the About box of Likno Web Modal Windows Builder, with your license information.

Properties

Part



5 Properties

5.1 Properties Overview

Properties are classified in the following categories:

The **Project Windows** properties that apply to each Modal Window of the loaded project separately:

- [Make links of your page behave as modal windows](#)
- [Sheets](#)
- [Content Source](#)
- [Content Text/HTML](#)
- [Page Link](#)
- [Image Link](#)
- [Page Element ID](#)
- [Modal Window Opens - When](#)

The **Project Style** properties which define the appearance and behavior of the Modal Windows of the loaded project:

- [Modal Window Opens - How](#)
- [Modal Window Closes - How](#)
- [Modal Window Opens - When](#)
- [Modal Window Closes - When](#)
- [Modal Window Z-Index](#)
- [Overlay Color](#)
- [Overlay Opacity](#)
- [Modal Window Position](#)
- [Modal Window CSS](#)
- [Modal Body CSS](#)
- [Modal Window Header](#)
- [Modal Window Footer](#)
- ["Loading" Info \(for AJAX only\)](#)

5.2 "Project Style" Properties

5.2.1 Modal Window Opens - How

Choose an opening effect for the modal window, along with its duration in milliseconds (ms).

The opening effect for the overlay is always fade-in.

See also

- [Property: Modal Window Closes - How](#)
- [Property: Modal Window Closes - When](#)
- [Property: Modal Window Opens - When](#)

5.2.2 Modal Window Opens - When

This property is window-specific only (not common) and is specified through the ["Opens When" property](#) of the "Project Windows" tab.

See also

["Project Windows": Opens When](#)
[Property: Modal Window Opens - How](#)
[Property: Modal Window Closes - How](#)
[Property: Modal Window Closes - When](#)

5.2.3 Modal Window Closes - How

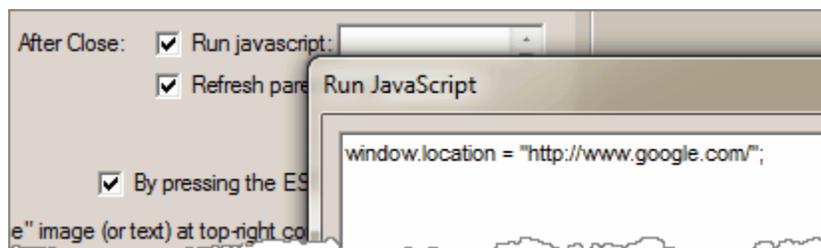
Choose a closing effect for the modal window, along with its duration in milliseconds (ms).

The closing effect for the overlay is always fade-out.

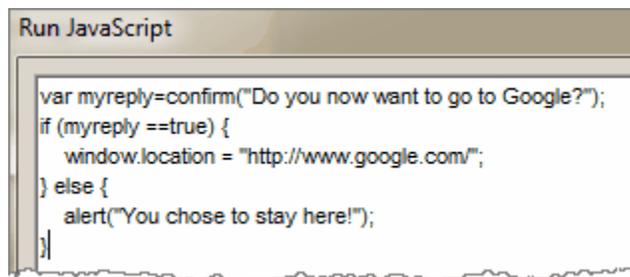
You can also choose the following extra actions after the modal window closes:

1. Run your own javascript code.

For example, this code transfers you to Google after the modal window closes:



Or you can use more lines of code, for advanced scripting:



Note that you should NOT use the `<script>` and `</script>` tags. Just enter your javascript code directly.

2. Refresh the page.

See also

[Property: Modal Window Opens - How](#)
[Property: Modal Window Closes - When](#)
[Property: Modal Window Opens - When](#)

5.2.4 Modal Window Closes - When

Choose when the modal window closes.

You have the following options:

- By clicking on the **overlay**
- By pressing the **ESC** key

Note! When interacting with an external website the focus is at the website so clicking the ESC key will function within the external page and not for the modal window. You will have to bring the focus into the main page (by clicking on the overlay) to gain the ESC key functionality back.

- By clicking on a special "**Close**" image (or text)

Here you can specify an image (or text) that closes the modal window. By default this "close" section is positioned at the top-right corner of the modal window.

Through the **Close HTML** field you customize the content of this section. The %CLOSE_IMAGE% variable represents the image specified in the "**Close Image**" field.

Through the **Close CSS** field you customize the look and positioning of this section.

- By clicking on any element that you add to your window's content with the attribute **class="<ProjectName>-close"** or by using the JavaScript command **<ProjectName>.close();** where <ProjectName> denotes the compile name you have given to your project.

Examples:

```
<a class="likno-modal-project-close" href="javascript:void(0)">Close this window</a>
```

```

```

```
<a onclick="likno_modal_project.close();" href="javascript:void(0)">Close this window</a>
```

```

```

How it works:

You can use this option to specify one or more **custom** areas (links, buttons, etc.) inside the modal window that close the window when clicked.

This option is always enabled.

If you do not add any element with that class attribute to your window content, the user will still be able to close the window through any of the other options (if enabled).

Finally, the JavaScript command can also take, as parameter, a function to execute once the modal window is closed.

Look at "Example 12" inside the application to see how this is used to close a modal window and open another one at the same time!

Note! If you use an "external page" as the source and you want to have a "close" button within this page then the above will not work! This is because "external pages" are displayed within an iFrame and hence do not have direct access to the Modal Window code which is linked to the parent frame. In this case you can use the "onclick" method, but instead of "likno_modal_project.close();" you should use **"parent.likno_modal_project.close();"**.

See also

- [Property: Modal Window Opens - How](#)
- [Property: Modal Window Closes - How](#)
- [Property: Modal Window Opens - When](#)

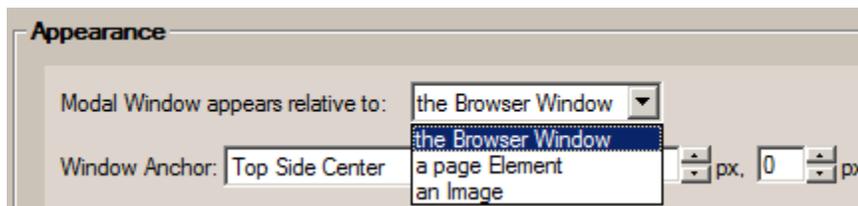
5.2.5 Modal Window Position

Select where the modal window opens on the web page.

You can match any corner (or side) of a "target" (which is the "Browser Window" or a "page Element" or an "Image") with any corner (or side) of the modal window.

You can also use the offsets to further refine your positioning relative to the selected "anchor" point.

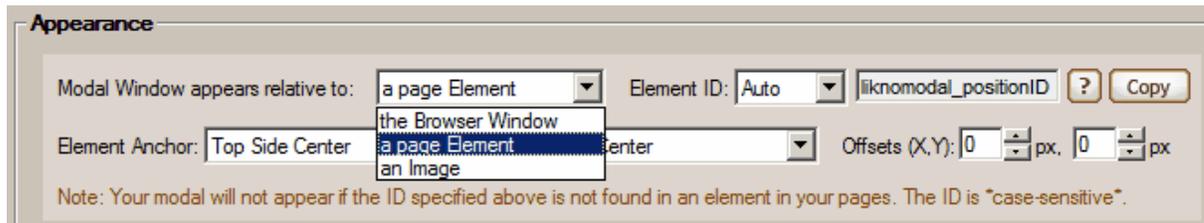
Option 1: Positioning the modal window relative to the browser's **window**



All you have to do is specify the Corner or Side of the browser window along with the Offsets X and Y and the modal window will appear at that distance from your selected window corner (or side).

Option 2: Positioning the modal window relative to an **element** of your pages (span,

div, td, table, etc.)



This option requires that you add extra code on the page but gives advanced flexibility (if needed).

How?

1. First make sure that the "**page Element**" option is selected at this property prior to the project's compilation (see screenshot above)
2. Edit your HTML page
3. Choose the element where you want your modal window to appear relative to
4. Add the following parameter on that element's tag:

```
id="liknomodal_positionID"
```

OR

```
id="<ModalName>_positionID"
```

(in case you choose to "customize" this property for a particular modal window, through the "Edit style of specific window" option at the top of the "Style Editor (F3)")

This is all you have to do. When a user browses your page, the Likno script detects the element with that id parameter and opens the modal window relative to the element's location. If the script does not detect the id, the modal window cannot appear.

You may also choose to use your own ID. 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 suggested ID to your selected element. In that case, you simply copy this "id" of your selected element and paste it as the **Element ID** in the application.

Examples:

The modal window is attached to a table using the following code:

```
<table id="liknomodal_positionID" 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="liknomodal_positionID"> &nbsp; </span>
```

TD:

```
<td id="liknomodal_positionID" valign="top">&nbsp;</td>
```

Link:

```
<a id="liknomodal_positionID" href="#">Modal Window attached to this link.</a>
```

Image: (while you can add the id to an image it is recommended to use the independent "Image" option instead, see below)

```

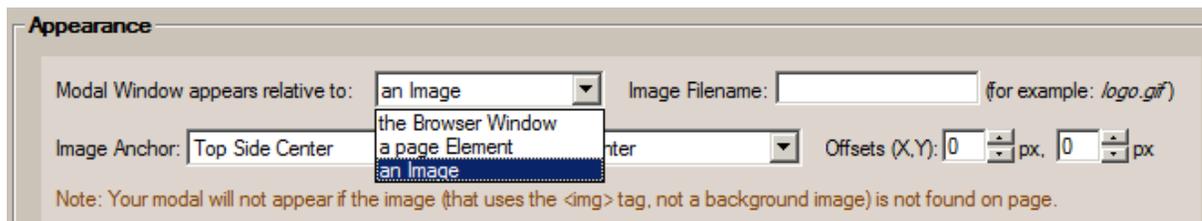
```

DIV:

```
<div id="liknomodal_positionID"> &nbsp;</div>
```

Note: You may also use the "class" attribute instead of the "id" attribute under the same concept. The script first tries to detect an "id" with the value that you specified, if it does not find one then it tries to detect a "class" with the same value. If multiple page elements use this "class", then the first one detected in the page will be used.

Option 3: Positioning the modal window relative to an **image** existent in your pages



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

You only specify the filename of the image (which should exist in your page!) and your positioning is complete: the modal window will appear relative to that image.

See also

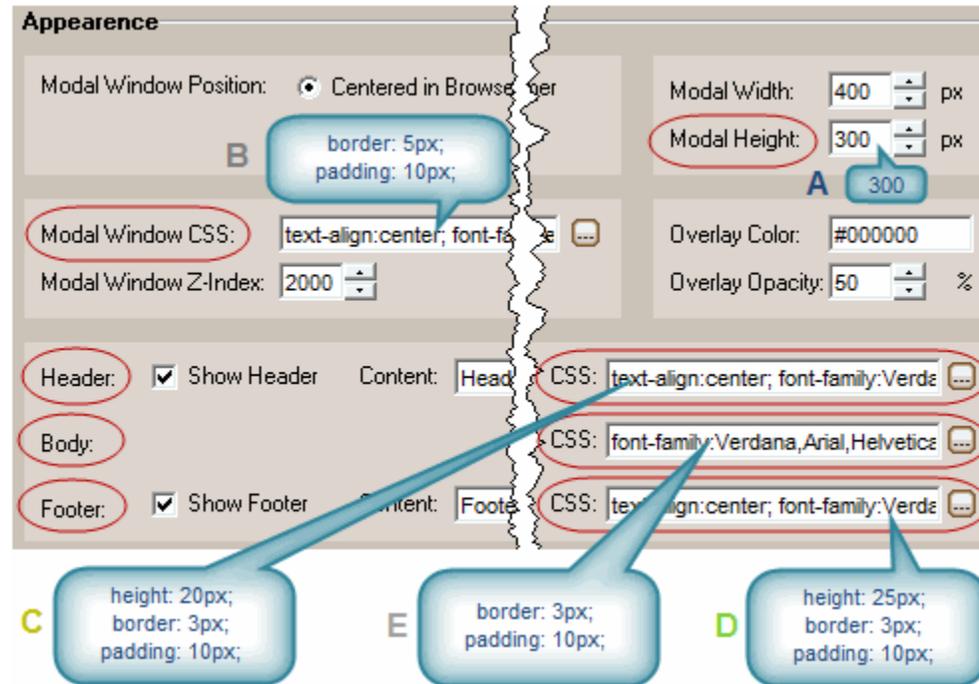
[Property: Modal Window CSS](#)

5.2.6 Modal Window Width/Height

Choose the dimensions (size) of the modal window.

Click the link below to read a blog post (with images), explaining about the 4 different areas and how their heights are specified:

<http://www.likno.com/blog/jquery-modal-window-heights/910/>



See also

[Property: Modal Window CSS](#)

[Property: Body CSS](#)

5.2.7 Modal Window CSS

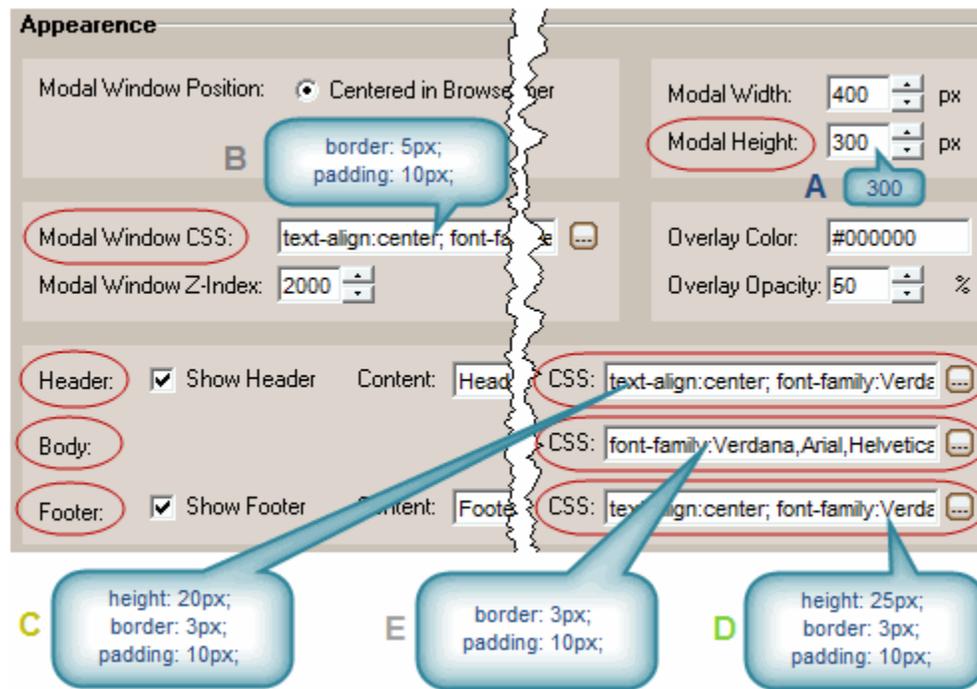
Modify the default CSS code to change the style of the modal window.

Change the value of this property to change the style of your modal window. The code used in this field is pure CSS code. You may also change only the "Body" area through the [<Body CSS>](#) property.

For more information about CSS styling, visit <http://www.likno.com/cssref>

Click the link below to read a blog post (with images), explaining about the 4 different areas and how their heights are specified:

<http://www.likno.com/blog/jquery-modal-window-heights/910/>



See also

- [Property: Body CSS](#)
- [Property: Modal Window Width/Height](#)
- [Property: Modal Window Position](#)

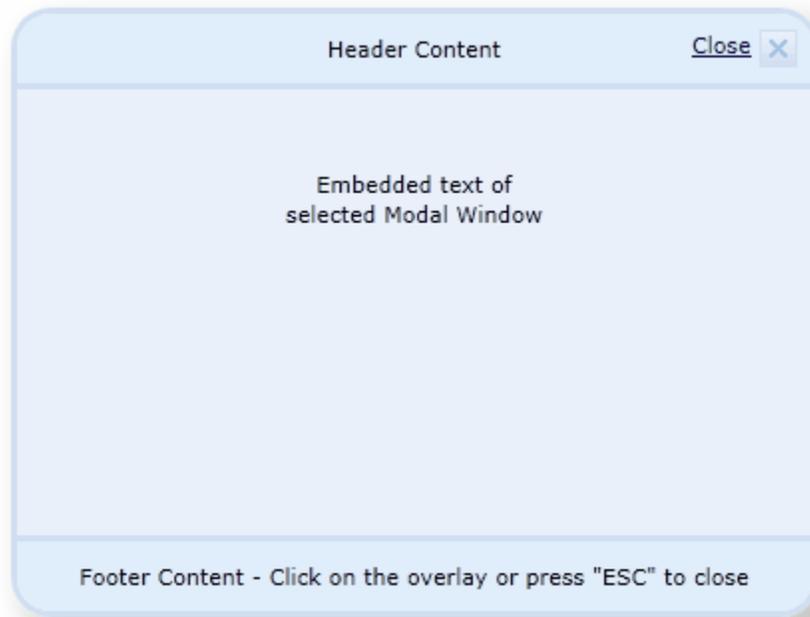
5.2.8 Modal Window Z-Index

Choose the Z-Index of all modal windows in the project.

Make sure that this value is higher than any other Z-Index value in your page so that the modal windows appear above all your page elements. This is why the default value is a high number (2000).

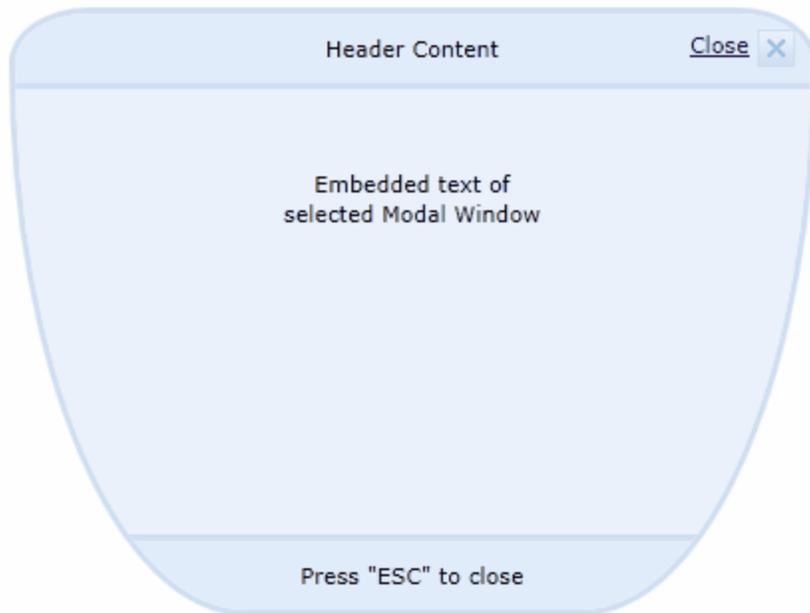
5.2.9 Modal Window Rounded Corners

Specifies if the corners of the window will look rounded and to which degree (through the X and Y radius values).



(a [Shadow](#) is also used above)

You can adjust the roundness of each corner all together or **even separately** (see example below).



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 CSS3** problem as the script degrades this feature gracefully on those browsers by **ignoring** its settings and showing all content.

Internet Explorer 9 also **requires** that your pages use a full DOCTYPE declaration, otherwise it cannot support the CSS3.

For example, the following pages will **not** show rounded corners in IE9...

```
1 <html>
2 <head>
3 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
2 <html>
3 <head>
4 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

...but the following pages **will** show rounded corners in IE9:

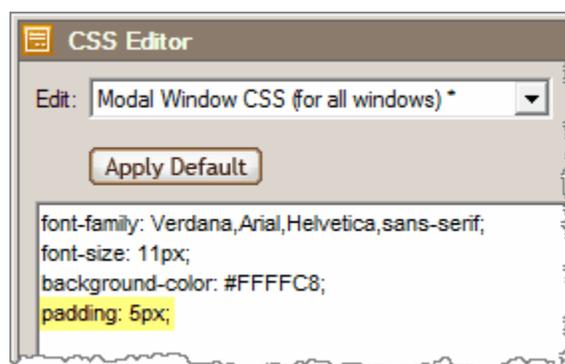
```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

You can find more info on the **border-radius** capabilities here:

<http://www.css3.info/preview/rounded-border/>

Certain window designs may also need some **padding** to show better in **Opera** and WebKit-based browsers, such as **Chrome, Safari**.

You can **add padding** through the [Modal Window CSS](#) property:

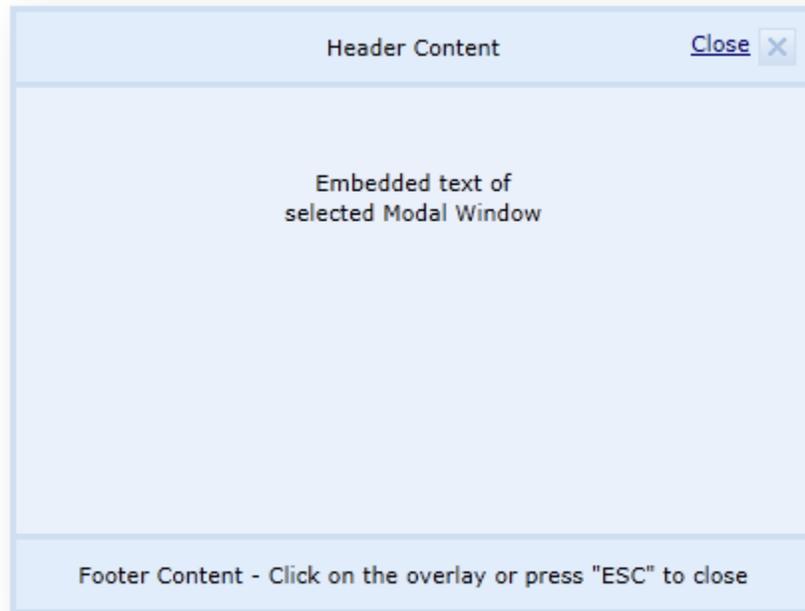


See also

[Modal Window CSS](#)

5.2.10 Modal Window Shadow

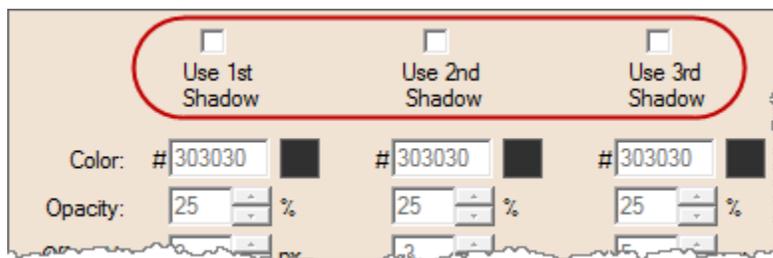
Specifies if the window has a shadow and how this shadow looks like.



You can specify several appearance options for a shadow:

color, opacity, offsets, blur, spread

You can also mix up to **3 different shadows into a single shadow** to create unique effects:



Note that when using a shadow you may also want to adjust the [Overlay Opacity](#) for better results.

Important! The "shadow" feature is based on the **box-shadow** property of the **CSS3** specification, which is available in **Firefox 3.5+, Safari (Webkit 522+), Chrome and others**. All earlier browsers (such as IE8 or below) **do not support** this feature. This is not a problem as the script degrades this feature gracefully on those browsers by **ignoring** its settings and showing the modal window without shadow.

Internet Explorer 9 also requires that your pages use a **full DOCTYPE declaration**, otherwise it cannot support the shadow feature.

property (see example below).

For example, the following pages will **not** show shadows in IE9...

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
2 <html>
3   <head>
4     <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

...but the following pages **will** show shadows in IE9:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
2 <html xmlns="http://www.w3.org/1999/xhtml">
3   <head>
4     <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/>

See also

[Overlay Opacity](#)

5.2.11 Overlay Color

Choose the color of the overlay that covers the screen.

See also

[Property: Overlay Opacity](#)

5.2.12 Overlay Opacity

Choose the opacity of the overlay that covers the screen.

Choose a value a value from 0 to 100. A lower value makes the overlay more transparent.

See also

[Property: Overlay Color](#)

5.2.13 Header

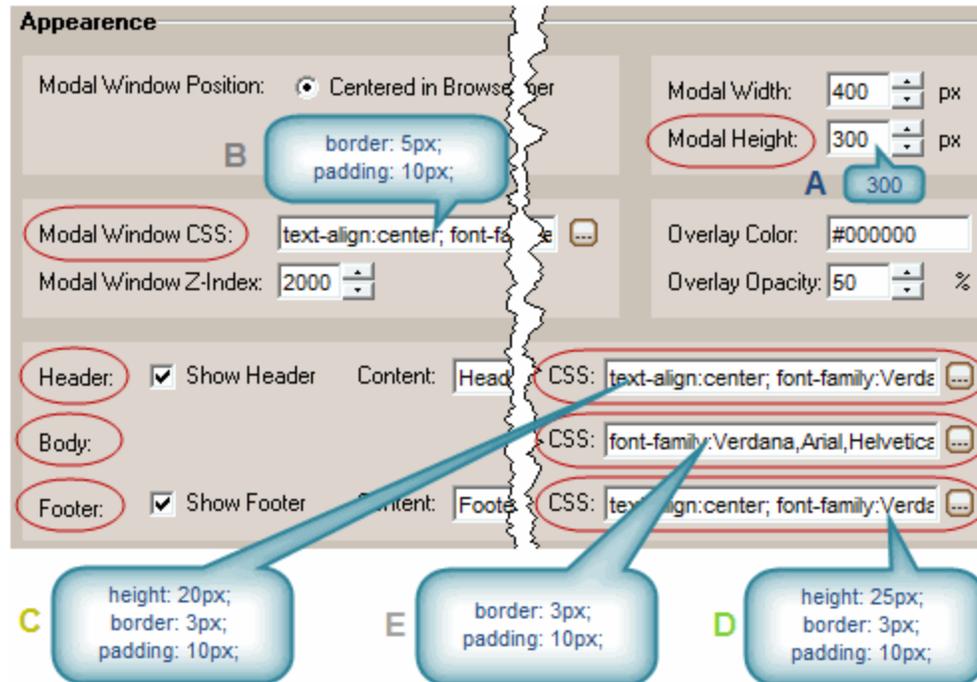
Specifies whether the modal window will show a Header or not.

If you select this property (check the box) then a Header will show up at the top of the modal window.

In the **<Content>** property define the **content** of the Header (HTML code accepted).
 In the **<CSS>** property define the **style** of the Header (only valid CSS code accepted).
 For more information about CSS styling, visit <http://www.likno.com/cssref>

Click the link below to read a blog post (with images), explaining about the 4 different areas and how their heights are specified:

<http://www.likno.com/blog/jquery-modal-window-heights/910/>



See also

[Property: Modal Window Footer](#)

5.2.14 Body CSS

Modify the default CSS code to change the style of the modal's Body.

The default value is the following:

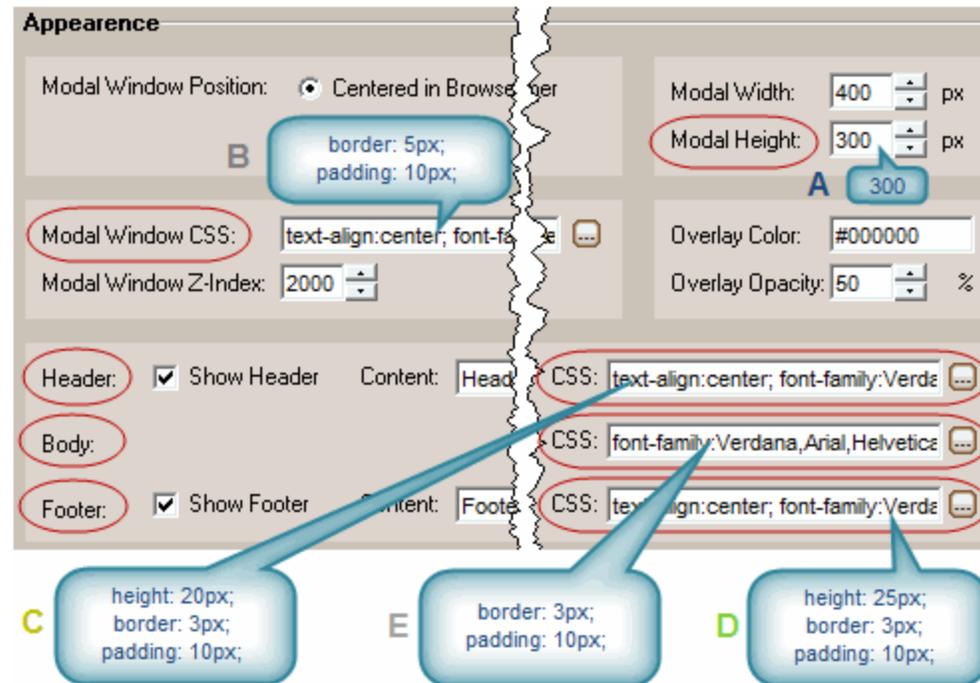
```
font-family:Verdana,Arial,Helvetica,sans-serif;
font-size:11px;
background-color:#EAF1FB;
border:3px solid #CFDEF1;
padding:3px;
```

Change the values of the above properties and add more properties to change the style of the Body of your modal window. The code used in this field is pure CSS code.

For more information about CSS styling, visit <http://www.likno.com/cssref>

Click the link below to read a blog post (with images), explaining about the 4 different areas and how their heights are specified:

<http://www.likno.com/blog/jquery-modal-window-heights/910/>



See also

[Property: Modal Window CSS](#)

[Property: Modal Window Width/Height](#)

[Property: Modal Window Position](#)

5.2.15 Footer

Specifies whether the modal window will show a Footer or not.

If you select this property (check the box) then a Footer will show up at the bottom of the modal window.

In the **<Content>** property define the **content** of the Footer (HTML code accepted).

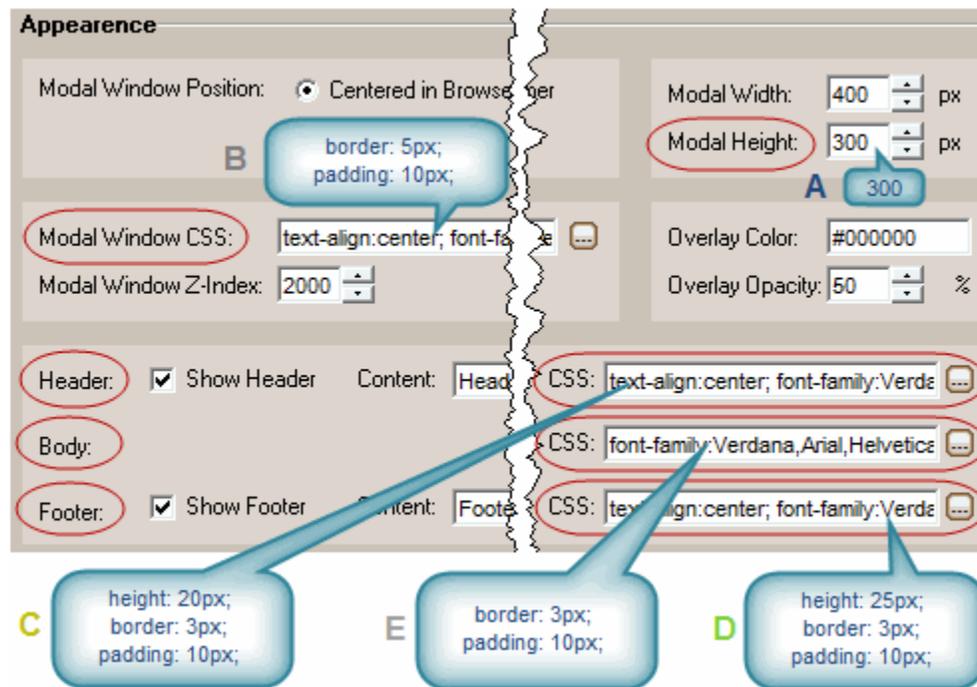
In the **<CSS>** property define the **style** of the Footer (only valid CSS code).

For more information about CSS styling, visit <http://www.likno.com/cssref>

Note: Do **not** use the **'height'** parameter in the Footer CSS. It will break your modal window if it is of the **multi-sheet** type.

Click the link below to read a blog post (with images), explaining about the 4 different areas and how their heights are specified:

<http://www.likno.com/blog/jquery-modal-window-heights/910/>

**See also**

[Property: Modal Window Header](#)

5.2.16 AJAX "Loading" Info

This option applies only when the content of the modal window is retrieved via an AJAX call (a call to another file located in your website).

Until the content is retrieved, an "AJAX loading image/text" will appear to prompt the viewer to wait. You can select among the included animated images within the application, or you can use your own image.

For more custom animated images visit <http://www.ajaxload.info>

See also

[Property: Page Link](#)

5.2.17 Multi-Sheet properties

This option applies only when the content of the modal window is of the "[multi-sheet](#)" type.

Navigation Bar:

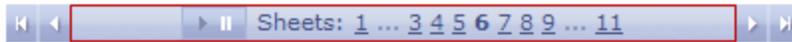
This set of properties allows you to choose one of the ready-made Navigation Bars and then fully customize it to match your needs.

Through the property "**Navigation Bar CSS Classes**" you may also edit the separate CSS for any of the following elements:

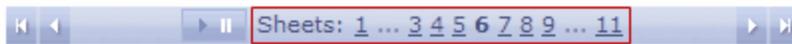
.navigationBarArea {}



.navigationBarCenterSection {}



.navigationBarSheetSection {}



.navigationBarFirst {}



.navigationBarPrev {}



.navigationBarNext {}



.navigationBarLast {}



.navigationBarPlayPause {}



.navigationBarText {}



.navigationBarSheetLink {}



.navigationBarSheetLinkHover {}
(CSS of the above element for the "mouse over" state)

.navigationBarSheetSelected {}



`.navigationBarSheetDots {}`



`.navigationBarSheetsSpacer {}`



Finally, you have 3 classes for extra styling of the "Bullets", if you choose to use them:

`.navigationBarBulletsNormal {}`
`.navigationBarBulletsOver {}`
`.navigationBarBulletsSelected {}`



Note that the `.navigationBarSheetsSpacer` also applies to Bullets:



Using the above CSS Classes you can change the look and feel of your Navigation Bar considerably!

This is an example on how different you can make the Navigation Bar by using your own images and CSS Classes:



Play Sheets:

Through these properties you can choose the following:

- Your SlideShow can **start automatically** on page load (or not).
- When the user puts the mouse on the Body area of the modal window (not on the header or footer area), then the SlideShow can pause.
- By increasing the "Delay" you can make your sheets (slides) show longer before the next sheet appears.
- Through the "Direction" property, you can make your sheets show in "random" order or even in "reverse" order (going backwards).

Other:

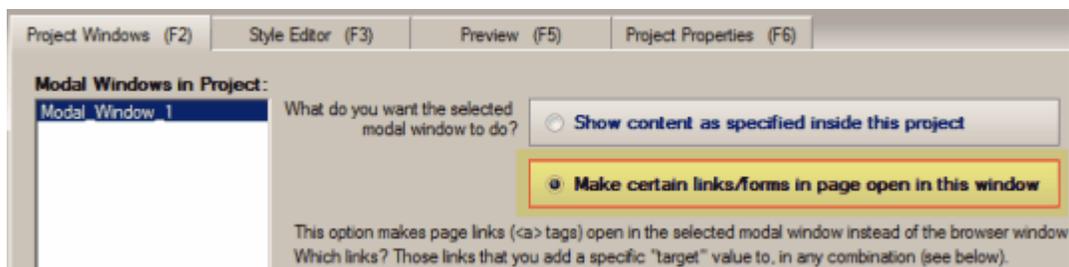
- The "Starting Sheet" can be different from "1" (the first sheet).
(Note: This value is ignored if you open the modal window with an opening sheet parameter. [see when](#))
- Through the "Loop" property you can specify whether the SlideShow will start over when reaching the final sheet (vice versa if "Direction" is "Reverse") or not.

Note: For more information about CSS styling, visit <http://www.likno.com/cssref>

5.3 "Project Windows" Properties

5.3.1 Make certain links/forms in a page open in this window

This is the 2nd main option in the "Project Windows" tab:



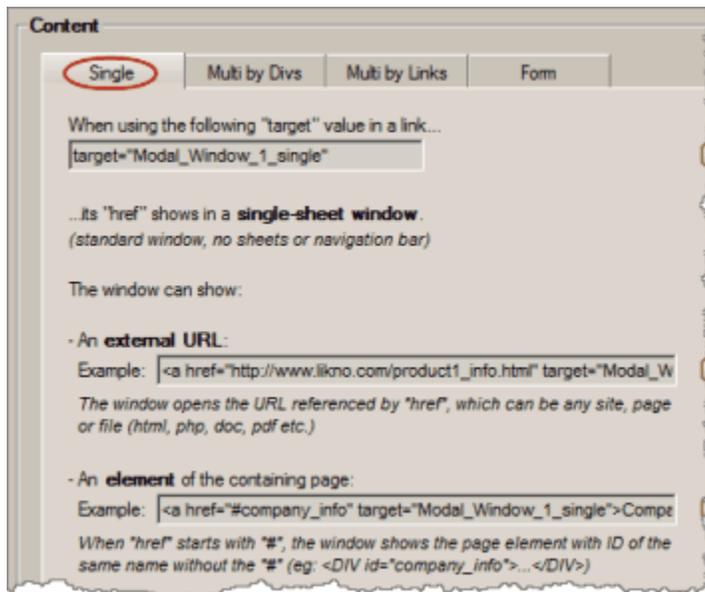
When using this option, the behavior of specified links (<a> tags) or forms (<form> tags) in your page changes: instead of opening in the browser window (when clicked) they open in the selected modal window.

But which links?

Only those page links that you add a **"target"** HTML attribute to, with a specific value.

This specific "target" value should be the **window's name** followed by one of the words **"_single"**, **"_multi_divs"** or **"_multi_links"** or **"_form"** which informs the modal window how to "use" that link (see below):

"Single" method



Examples:

```
<a href="http://www.likno.com/product1_info.html" target="Modal_Window_1_single">Info on Product 1</a>
```

```
<a href="#company_info" target="Modal_Window_1_single">Company Info</a>
```

When using the **_single** suffix, clicking on the link **opens** the modal window as **"single-sheet"** (standard window, without a Navigation Bar).

The content shown is specified through the **href** attribute:

1. Content is an **external URL**:

The window shows content located outside the containing page, such as another html page, an image, pdf file etc.

```
<a href="http://www.likno.com/product1_info.html" target="Modal_Window_1_single">Info on Product 1</a>
```

```
<a href="http://www.google.com" target="Modal_Window_1_single">Open Google</a>
```

```
<a href="http://www.likno.com/Images/logo.gif" target="Modal_Window_1_single">View Image</a>
```

```
<a href="http://www.likno.com/download/AllWebMenus_Manual.pdf" target="Modal_Window_1_single">View PDF</a>
```

If the URL (a page, image, etc.) is located inside your website, you may also use a "document-relative" or "site-relative" path. If you use a "document-relative" path, just make

sure that this is relative to the page containing (showing) the modal window.

Using "document-relative" path:

```
<a href="product1_info.html" target="Modal_Window_1_single">Info on Product 1</a>
```

```
<a href="Images/logo.gif" target="Modal_Window_1_single">View Image</a>
```

or (if page is located in a subfolder)

```
<a href="../product2_info.html" target="Modal_Window_1_single">Info on Product 2</a>
```

```
<a href="../Images/logo.gif" target="Modal_Window_1_single">View Image</a>
```

Using "site-relative" path:

```
<a href="/product1_info.html" target="Modal_Window_1_single">Info on Product 1</a>
```

```
<a href="/Images/logo.gif" target="Modal_Window_1_single">View Image</a>
```

2. Content is an **element** (DIV, IMG etc.) of the containing page:

The window shows an element located inside the containing page. Which element? The one that uses the same ID as the one used in the href attribute, without the "#".

For example, for this <a> link...

```
<a href="#company_info" target="Modal_Window_1_single">Company Info</a>
```

...the modal window shows the following "in-page" element that uses this ID:

```
<div id="company_info" style="display:none; text-align: center;">
  <p>&nbsp;</p>
  <p>Hello!</p>
  <p>&nbsp;</p>
  <p>Our company is here to serve you!</p>
</div>
```

Notes:

- You are not restricted to a DIV, you can show any type of element, like an image for example:

```

```

- The ID is always unique per element, you will get an error if you use the same ID on another page element.

- Once the modal window appears the element disappears from your page afterwards. Therefore make sure that your page element is not a visible element, but it is hidden through the *style="display: none;"* code.

- Keep in mind that if the element you specified is **not** in your page the modal window will **not**

appear. Make sure that you have correctly selected which element will appear by adding the correct ID attribute.

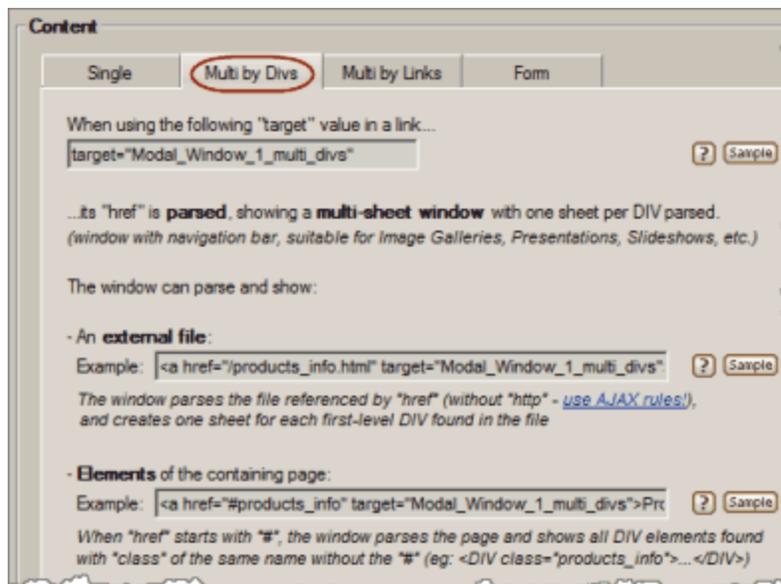
Global note: You may also add the following optional parameters to change the specified modal's width, height, header content or footer content:

```
data-lwmw-width="400px" (or "70%", or "auto")
data-lwmw-height="300px" (or "70%", or "auto")
data-lwmw-header="your custom header content here"
data-lwmw-footer="your custom footer content here"
```

This is an example:

```
<a href="http://www.likno.com/product1_info.html" target="Modal_Window_1_single"
data-lwmw-width="70%" data-lwmw-height="auto" data-lwmw-header="This is a new HEADER!"
data-lwmw-footer="This is a new FOOTER!">Info on Product 1</a>
```

"Multi by Divs" method



This use case applies when you use the `_multi_divs` suffix next the window's name inside the "target" attribute of a page link:

Examples:

```
<a href="http://www.mysite.com/products_info.html" target="Modal_Window_1_multi_divs"
>Products Info</a>
```

```
<a href="#products_info" target="Modal_Window_1_multi_divs">Products Info</a>
```

When using the `_multi_divs` suffix, clicking on the link **parses its "href"** and opens a **multi-sheet window** with one **sheet per first-level DIV** found in the parsed HTML.

The HTML to be parsed is either a file or the page itself (the one that opens the window), and this is determined by the link's **href** value:

1. "href" references an **external file** (within the same site):

A **multi-sheet** window opens and populates its sheets "on-the-fly", by retrieving and parsing specific HTML from a file located on the same site.

The HTML contained in the file should follow the **"one-DIV-per-sheet" rule**, as described in the ["Embedded"](#) option (click this link to view how to structure the file!).

This option is useful to create and show database-driven or "on-the-fly" sheets, with updated content each time the modal window opens (as long as your PHP or ASP.NET, etc. refreshes this file before the modal window accesses it).

Examples:

Using "document-relative" path:

```
<a href="products_info.html" target="Modal_Window_1_multi_divs">Products Info</a>
```

or (if page is located in a subfolder)

```
<a href="../products_info.html" target="Modal_Window_1_multi_divs">Products Info</a>
```

Using "site-relative" path:

```
<a href="/product1_info.html" target="Modal_Window_1_multi_divs">Products Info</a>
```

Notes:

- The path of the file should be **relative** to the location of the page that opens the modal window; you **cannot** use absolute URLs, such as http:// or https://
- The specified file should be present on your website, the **Preview** cannot show it!

2. "href" starts with **"#"** (to indicate parsing of **page elements**):

A **multi-sheet** window opens and populates its sheets with **elements (DIVs, etc.) contained in the same page** that contains the window.

But which elements?

Only those elements that you add a specific **"class"** value to. This specific "class" value should be the **window's name** (without any "#").

For example, for this <a> link...

```
<a href="#products_info" target="Modal_Window_1_multi_divs">Products Info</a>
```

...the modal window shows the following "in-page" elements (one sheet per DIV), that use this "class":

```
<div class="products_info" style="text-align:center; display: none;">
  <p>&nbsp;</p>
  <p>Modal Window for<br><br><b>AllWebMenus</b></p>
  <p>Create javascript menus for<br>websites in just a few clicks!</p>
  <p><a href="http://www.likno.com/allwebmenusinfo.html" target="_top">more info on this
  product</a></p>
</div>

<div class="products_info" style="text-align:center; display: none;">
  <p>&nbsp;</p>
  <p>Modal Window for<br><br><b>Likno Web Button Maker</b></p>
  <p>Create professional and stylish<br>web buttons in just a few clicks!</p>
  <p><a href="http://www.likno.com/web-button-maker/index.html" target="_top">more info
  on this product</a></p>
</div>

<div class="products_info" style="text-align:center; display: none;">
  <p>&nbsp;</p>
  <p>Modal Window for<br><br><b>Photo Frame Show</b></p>
  
  <p>Framed photo created with our Photo Frame Show software<br><a
  href="http://www.frameshow.com">check it out!</a></p>
</div>
```

The above html populates a modal window with 3 sheets. The content of each sheet is the HTML within each element (DIV). Note that you are not limited to the <div> element, you can use any other element you like (like <p>, , , etc.), as long as such elements use the same class value.

Notes:

- Once the modal window appears the elements disappear from your page afterwards. Therefore make sure that your page elements are not visible elements, but hidden through the `style="display: none;"` code.
- Keep in mind that if **no** elements are found in your page with that class the modal window will **not appear**. Make sure that you have correctly selected which elements will appear by adding the correct class attribute.
- If your element already uses a class value, HTML allows you to use **multiple class values** separated by space:

```
class="anotherClass products_info"
```

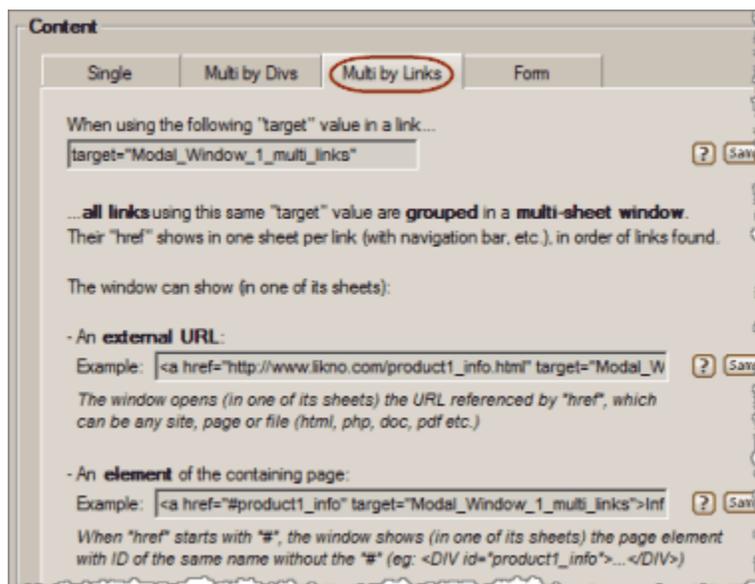
Global note: You may also add the following optional parameters to change the specified modal's width, height, header content or footer content:

```
data-lwmw-width="400px" (or "70%", or "auto")
data-lwmw-height="300px" (or "70%", or "auto")
data-lwmw-header="your custom header content here"
data-lwmw-footer="your custom footer content here"
```

This is an example:

```
<a href="http://www.likno.com/product1_info.html" target="Modal_Window_1_multi_divs"
data-lwmw-width="70%" data-lwmw-height="auto" data-lwmw-header="This is a new HEADER!"
data-lwmw-footer="This is a new FOOTER!">Info on Product 1</a>
```

"Multi by Links" method



This use case applies when you use the `_multi_links` suffix next the window's name inside the "target" attribute of a page link:

Examples of such links in the HTML of a page:

<!-- using "external URL" in "href" for external content source -->

```
<a href="http://www.likno.com/product1_info.html" target="Modal_Window_1_multi_links"
>Info on Product 1</a>
```

```
<a href="http://www.likno.com/product2_info.html" target="Modal_Window_1_multi_links"
>Info on Product 2</a>
```

```
<a href="http://www.likno.com/product3_info.html" target="Modal_Window_1_multi_links"
>Info on Product 3</a>
```

```
<a href="http://www.likno.com/product4_info.html" target="Modal_Window_1_multi_links"
>Info on Product 4</a>
```

<!-- using "#" in "href" for "in-page" element content source -->

```
<a href="#product1_info" target="Modal_Window_2_multi_links">Info on Product 1</a>
```

```

<a href="#product2_info" target="Modal_Window_2_multi_links">Info on Product 2</a>
<a href="#product3_info" target="Modal_Window_2_multi_links">Info on Product 3</a>
<a href="#product4_info" target="Modal_Window_2_multi_links">Info on Product 4</a>

<!-- using both content sources! -->

<a href="http://www.likno.com/product1_info.html" target="Modal_Window_3_multi_links"
>Info on Product 1</a>

<a href="#product2_info" target="Modal_Window_3_multi_links">Info on Product 2</a>
<a href="#product3_info" target="Modal_Window_3_multi_links">Info on Product 3</a>

<a href="http://www.likno.com/product4_info.html" target="Modal_Window_3_multi_links"
>Info on Product 4</a>

```

(these links open 3 differently designed modal windows, showing 4 sheets each)

When using the **_multi_links** suffix, clicking on the link **opens** the modal window as **"multi-sheet"** (with navigation bar, etc), with other sheets being populated by all links of the same page that use the same "target" value (with the "_multi_links" suffix on them).

It is actually the same as the "_single" method (see above), with the only difference being that the window shows other sheets too, by **"grouping"** all similar links (the ones with same "target" value).

As with the "_single" method, the content shown (in the opening sheet of the multi-sheet window) is specified through the **href** attribute:

1. Content is an **external URL**:

The window shows content located outside the containing page, such as another html page, an image, pdf file etc.

```

<a href="http://www.likno.com/product1_info.html" target="Modal_Window_1_multi_links">Info
on Product 1</a>

<a href="http://www.google.com" target="Modal_Window_1_multi_links">Open Google</a>

<a href="http://www.likno.com/Images/logo.gif" target="Modal_Window_1_multi_links">View
Image</a>

<a href="http://www.likno.com/download/AllWebMenus_Manual.pdf"
target="Modal_Window_1_multi_links">View PDF</a>

```

If the URL (a page, image, etc.) is located inside your website, you may also use a "document-relative" or "site-relative" path. If you use a "document-relative" path, just make sure that this is relative to the page containing (showing) the modal window.

Using "document-relative" path:

```

<a href="product1_info.html" target="Modal_Window_1_multi_links">Info on Product 1</a>

```

```
<a href="Images/logo.gif" target="Modal_Window_1_multi_links">View Image</a>
```

or (if page is located in a subfolder)

```
<a href="../product2_info.html" target="Modal_Window_1_multi_links">Info on Product 2</a>
```

```
<a href="../Images/logo.gif" target="Modal_Window_1_multi_links">View Image</a>
```

Using "site-relative" path:

```
<a href="/product1_info.html" target="Modal_Window_1_multi_links">Info on Product 1</a>
```

```
<a href="/Images/logo.gif" target="Modal_Window_1_multi_links">View Image</a>
```

2. Content is an **element** (DIV, IMG etc.) of the containing page:

The window shows an element located inside the containing page.

Which element?

The one that uses the same ID as the one used in the href attribute, without the "#".

For example, for this <a> link...

```
<a href="#company_info" target="Modal_Window_1_multi_links">Company Info</a>
```

...the modal window shows the following "in-page" element that uses this ID:

```
<div id="company_info" style="display:none; text-align: center;">
  <p>&nbsp;</p>
  <p>Hello!</p>
  <p>&nbsp;</p>
  <p>Our company is here to serve you!</p>
</div>
```

Notes:

- You are not restricted to a DIV, you can show any type of element, like an image for example:

```

```

- The ID is always unique per element, you will get an error if you use the same ID on another page element.

- Once the modal window appears the element disappears from your page afterwards. Therefore make sure that your page element is not a visible element, but it is hidden through the `style="display: none;"` code.

- Keep in mind that if the element you specified is **not** in your page the modal window will **not appear**. Make sure that you have correctly selected which element will appear by adding the correct ID attribute.

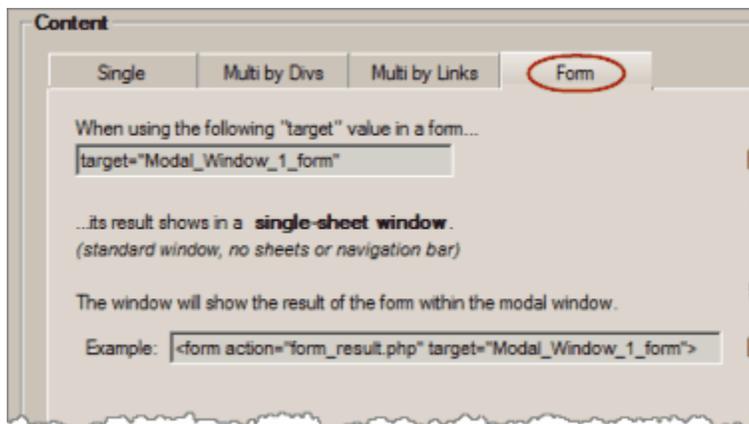
Global note: You may also add the following optional parameters to change the specified modal's width, height, header content or footer content:

```
data-lwmw-width="400px" (or "70%", or "auto")
data-lwmw-height="300px" (or "70%", or "auto")
data-lwmw-header="your custom header content here"
data-lwmw-footer="your custom footer content here"
```

This is an example:

```
<a href="http://www.likno.com/product1_info.html" target="Modal_Window_1_multi_links"
data-lwmw-width="70%" data-lwmw-height="auto" data-lwmw-header="This is a new HEADER!"
data-lwmw-footer="This is a new FOOTER!">Info on Product 1</a>
```

"Form" method (for page forms, i.e. <form> tags, not links)



Contrary to the other 3 methods, this method is not related to page links (<a> tags) but to **page forms (<form> tags)**.

It is used so that the user views the results of a form (i.e. the content after the form's "submit" button is pressed) into a modal window opening above the page.

This use case applies when you use the **_form** suffix next the window's name inside the "target" attribute of a page's <form> element:

Examples of such forms in the HTML of a page:

```
<!-- FORM 1 -->
```

```
<form action="http://www.likno.com/form1.php" method="POST"
target="Modal_Window_1_form">
```

```
What is your name?<br>
```

```

<table class="highlightText" style="width: 220px; border: none;">
  <tr><td> Name: </td><td> <input type="text" name="user_name"/> </td></tr>
  <tr><td> Surname: </td><td> <input type="text" name="user_surname"/> </td></tr>
  <tr><td colspan="2" align="center"><input type="submit" value="Submit"></td></tr>
</table>
</form>

<!-- FORM 2 -->

<form action="http://www.likno.com/form2.php" method="POST"
target="Modal_Window_1_form">
  How would you like to travel today?<br>
  <input type="radio" name="transport" id="car" value="car" checked="checked"/><label
for="car">by car</label>
  <input type="radio" name="transport" id="bus" value="bus"/><label for="bus">by bus</label>
  <input type="radio" name="transport" id="bike" value="bike"/><label for="bike">by bike</label>
  <input type="submit" value="Select">
</form>

```

Global note: You may also add the following optional parameters to change the specified modal's width, height, header content or footer content:

```

data-lwmw-width="400px" (or "70%", or "auto")
data-lwmw-height="300px" (or "70%", or "auto")
data-lwmw-header="your custom header content here"
data-lwmw-footer="your custom footer content here"

```

This is an example:

```

<form action="http://www.likno.com/form2.php" method="POST" target="Modal_Window_1_form"
data-lwmw-width="70%" data-lwmw-height="auto" data-lwmw-header="This is a new HEADER!"
data-lwmw-footer="This is a new FOOTER!">

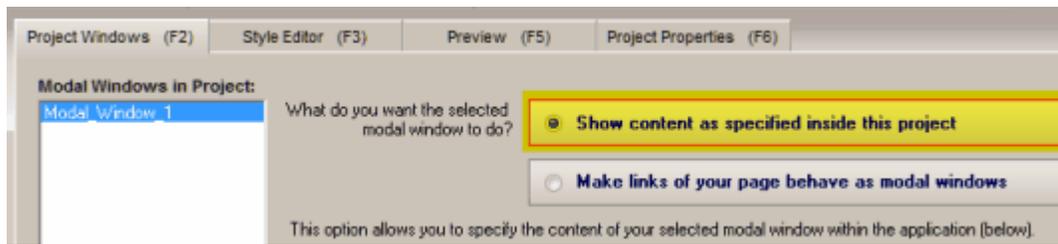
```

See also

[Show Content as specified inside the project Modal Windows opened from Frames or IFrames](#)

5.3.2 Show Content as specified inside this project

This is the 1st main option in the "Project Windows" tab:



When using this option, you specify the content (or links to the content) of your selected modal window within the application.

[More info on its various Content options](#)

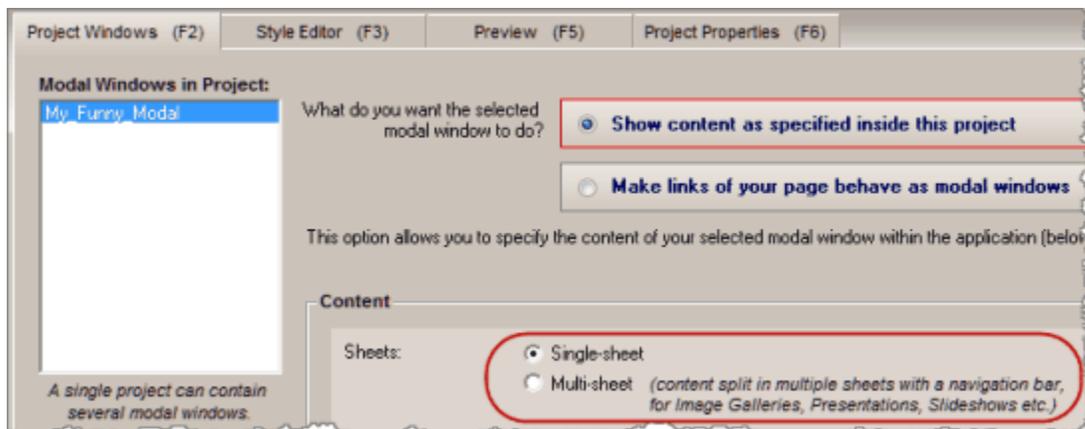
[More info on its various "Opens When" options](#)

See also

[Make links of your page behave as modal windows](#)

5.3.2.1 Sheets

Choose if the modal window will consist of a **single** sheet or **multiple** sheets:



- **Single-Sheet**

The modal window consists of one sheet.

Its <Content Source> property has the following options for defining the content of the modal window:

[Embedded Page Element](#)
[Image](#)
[AJAX](#)
[External Site](#)

- **Multi-Sheet (for Image Gallery, Presentations, SlideShows, etc.)**

The modal window consists of a number of sheets which are accessed through a [Navigation Bar](#) (optional) added before the footer.

Its <Content Source> property has the following options for defining the number of sheets and their content:

[Embedded](#)
[Page elements](#)
[HTML File](#)

5.3.2.2 Content Source options for a single-sheet window

See the following options on how to specify the content of a single sheet in a modal window:

[Embedded](#)
[Page Element](#)
[Image](#)
[AJAX](#)
[External](#)

See also:

[Content Source options for a multi-sheet window](#)

5.3.2.2.1 Embedded

Add the content of the modal window directly into the "Text/HTML" property.

You can either add simple text to be displayed, or more complex HTML code - including forms, different styling, flash movies, videos (eg: YouTube), etc.

Notes:

- If you use **"document-relative" paths** to reference pages or files (eg: `href="mypage.html"` or `src="image1.png"`), these should be relative to the page that contains and opens the modal window. [read more](#)

- For the Preview Panel to be able to show images using Site_Root-relative paths (example: `"/image1.gif"`) or Page-relative paths (example: `"image1.gif"` or `"pics/image1.gif"`), you need to add them to the "Project Assets" folder or to any other local folder that you specify through the **Image Folders for Preview Panel** list in [Preview Panel Settings](#).

- If you want to embed a **YouTube video** into a modal window just make sure that you are using the new embed style (since July 2010) that uses `<iframe>` instead of the old embed style that uses the `<object>` element (now obsolete).

See also

[How do I display a YouTube video in a Modal Window?](#)
[Content options for a single-sheet window](#)

5.3.2.2.2 Page Element

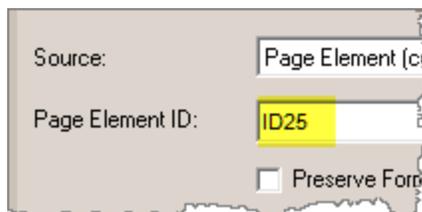
The content of the modal window will be the content of an element that is already (hidden) in your page.

Enter the value of the ID attribute of that element in the "Page Element ID" property. Make sure that the element has an **ID** attribute and it is **unique in the specific page**.

For example let's assume that you have the following element in your page:

```
<div id="ID25" style="display: none; padding: 5px; background-color:green;" class="myClass">  
    ...content of the element...  
</div>
```

In order to have this element open as a modal window, enter the **ID25** value in the <Page Element ID> field.



In case this element is a **form** and you wish to preserve its values between successive openings and closings of its modal window, you also need to check the "Preserve Form Values" option.

Important notes:

- Once the modal window appears the element disappears from your page afterwards. Therefore make sure that your page element is not a visible element, but it is hidden through the `style="display: none;"` code.
- Keep in mind that if the element you specified is **not** in your page the modal window will **not appear**. Make sure that you have correctly selected which element will appear by adding the correct id attribute.

See also

[Content options for a single-sheet window](#)

5.3.2.2.3 Image

Displays an image.

The image can be sourced either from the project's "assets" folder or from a web location (relative or absolute to the loaded page).

See also

[Content options for a single-sheet window](#)

5.3.2.2.4 AJAX

The content of the modal window is contained in a page, where that page follows the AJAX rules below:

1. Page is referenced with a **document-relative** path (relative to the location of the page that opens the modal window, see examples). You **cannot use absolute URLs**, such as "http://" or "https://". The page is **located in your domain** (website), not outside.

Examples: ajax1.php
 ajax2.htm
 /ajaxpages/ajax1.php

NOT: http://www.mysite.com/ajax1.php

2. Page **does NOT contain Javascript code.**

If your page contains javascript code or you want to reference it with an absolute URL, choose the "[External](#)" option instead.

The page should be present on your website, the **Preview Pane will not be able to show it** but your actual online page that uses the modal window will.

See also

[Content options for a single-sheet window](#)

5.3.2.2.5 External

The content of the modal window can be any site, page or file that can be loaded in a browser.

Add the path/name of the page (or file) in the "Link" property.

Examples: http://www.google.com/ (absolute path)
 my_page.html
 ../my_other_page.asp
 /docs/my_document.pdf

Notes if you use a relative path:

- The specified page should be present on your website, the **Preview** cannot show it!
- If you use a "document-relative" path to reference the page or file, this should be relative to the page that contains and opens the modal window.

See also

[Content options for a single-sheet window](#)

5.3.2.3 Content Source options for a multi-sheet window

See the following options on how to specify the content of multiple sheets in a modal window:

[Embedded](#)
[Page elements](#)
[HTML File](#)

See also:

[Content Source options for a single-sheet window](#)

5.3.2.3.1 Embedded

Create a **multi-sheet** modal window by adding the content of the modal window sheets **directly** into the "HTML" property of this project (i.e. content is embedded to the project).

How it works:

The HTML for each sheet is enclosed inside each **first-level DIV element**. You can use as many DIV elements as you wish. Each first-level DIV element defines one sheet.

Example:

Look at the following HTML that creates a 5-sheet modal window; the code for each sheet has a different color:

```
<div>
  <p>This is the 1st sheet.</p>
</div>

<div>
  <!-- This is the 2nd sheet and shows an image -->
  
</div>

<div>
  <!-- This is the 3rd sheet and shows content in divs and a paragraph -->
  <div style="text-align:center;">
    This is the first section of the 3rd sheet. It also contains an input area: <input type="button"
    value="button"></input>
  </div>
  <p>This example shows that you can have NESTED divs (or other tags) within the same
  SHEET.</p>
  <div style="text-align:justify;">
    This is the last section of the 3rd sheet. Internal DIVs (inside other DIVs) do <b>not</b> create
    separate sheets. Only the <b>first-level</b> DIVs count for creating a sheet.
  </div>
</div>

<div ajaxurl="mypages/ajaxpage.html">
  <!-- This is the 4th sheet, its content comes from another file, through AJAX -->
</div>

<div externalurl="http://www.google.com/">
  <!-- This is the 5th sheet, its content comes from an external site, displayed as an iframe -->
</div>
```

Note that each sheet is an element at the top level (it has no parent element).

- If you want a sheet to show **simple text** (sheet 1 in the example above) you need to add the text within a **div** tag.
- If you want a sheet to show an **image** (sheet 2 in the example above) you need to define the **img** tag inside a **div** tag.
- If you want a sheet to show **HTML content** (sheet 3 in the example above) you need to add the HTML code within a **div** tag.
- If you want a sheet to load **content from a page via an AJAX call** (sheet 4 in the example above) you need to define a **div** tag which will contain the attribute **ajaxurl**. The value of the **ajaxurl** attribute should be the web page that you want to show (relative to the location of the web page that opens the modal window).
- If you want a sheet to load **an external site as an iframe** (sheet 5 in the example above) you need to define a **div** tag which will contain the attribute **externalurl**. The value of the **externalurl** attribute should be the web page that you want to show (with the full URL).
- You may also add the following optional parameters on each **div** tag to change the specified modal's width, height, header content or footer content:

```
data-lwmw-width="400px" (or "70%", or "auto")
data-lwmw-height="300px" (or "70%", or "auto")
data-lwmw-header="your custom header content here"
data-lwmw-footer="your custom footer content here"
```

This is an example:

```
<div data-lwmw-width="70%" data-lwmw-height="auto" data-lwmw-header="This is a custom
HEADER!" data-lwmw-footer="This is a custom FOOTER!">
<p>This is the 1st sheet.</p>
</div>
```

Notes:

- If you use **"document-relative" paths** to reference pages or files (eg: `href="mypage.html"` or `src="image1.png"`), these should be relative to the page that contains and opens the modal window. [read more](#)
- For the Preview Panel to be able to show images using Site_Root-relative paths (example: `"/image1.gif"`) or Page-relative paths (example: `"image1.gif"` or `"pics/image1.gif"`), you need to add them to the "Project Assets" folder or to any other local folder that you specify through the **Image Folders for Preview Panel** list in [Preview Panel Settings](#).
- If you want to embed a **YouTube video** into a modal window just make sure that you are using the new embed style (since July 2010) that uses `<iframe>` instead of the old embed style that uses the `<object>` element (now obsolete).

See also

[Content options for a multi-sheet window](#)

5.3.2.3.2 Page elements

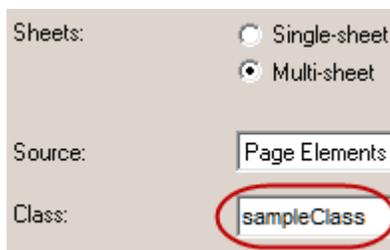
Create a **multi-sheet** modal window that populates its sheets through **page elements (DIV, etc.)** that use the **same class** (one sheet per element, in order of elements found).

- Example of HTML in your page:

```
<p>Page content here...</p>
<!-- The following DIVs show in sheets when the modal window opens, one sheet per DIV -->
<div class="sampleClass" style="text-align:center; display: none;">
  <p>&nbsp;</p>
  <p>Embedded text of<br>1st sheet</p>
  <p>&nbsp;</p>
  <p>Hello!</p>
</div>
<div class="sampleClass" style="text-align:center; display: none;">
  <p>&nbsp;</p>
  <p>Embedded text of<br>2nd sheet</p>
  <p>&nbsp;</p>
  <p>How are you?</p>
</div>
<div class="sampleClass" style="text-align:center; display: none;">
  
  <p>Framed photo created with our Photo Frame Show software<br><a
  href="http://www.frameshow.com">check it out!</a></p>
</div>
<div class="sampleClass" style="text-align:center; display: none;">
  <p>&nbsp;</p>
  <p>Embedded text of<br>4th sheet</p>
  <p>&nbsp;</p>
  <p>Let us know what to improve, thanks!<br><br>
  <i>(use the 'Send Feedback' button in the application)</i>
  </p>
</div>
<p>Page content continues...</p>
```

The above DIVs populate a modal window with 4 sheets. Each sheet's content is the HTML contained in each DIV. Note that you are not limited to the DIV element, you can also use any other element you like (like <p>, , , etc.), as long as such elements also use the same class value.

- In the application:
Enter the class value you chose (eg: *sampleClass*) in the **Class** property.



Important notes:

- "sampleClass" is an example, you can use any other value you like.

- If an element already uses another class value, HTML allows you to use **multiple class values** separated by space:

```
<div class="anotherClass sampleClass" style="display: none;">
```

- Once the modal window appears the elements disappear from your page afterwards. Therefore make sure that your page elements are not visible elements, but hidden through the `style="display: none;"` code.

- Keep in mind that if **no** elements are found in your page with that class the modal window will **not appear**. Make sure that you have correctly selected which elements will appear by adding the correct class attribute.

- You may also add the following optional parameters on each **div** tag to change the specified modal's width, height, header content or footer content:

```
data-lwmw-width="400px" (or "70%", or "auto")
data-lwmw-height="300px" (or "70%", or "auto")
data-lwmw-header="your custom header content here"
data-lwmw-footer="your custom footer content here"
```

This is an example:

```
<div class="sampleClass" style="text-align:center; display: none;" data-lwmw-width="70%"
data-lwmw-height="auto" data-lwmw-header="This is a custom HEADER!"
data-lwmw-footer="This is a custom FOOTER!">
<p>This is the 1st sheet.</p>
</div>
```

See also

[Content options for a multi-sheet window](#)

5.3.2.3.3 HTML File

Create a **multi-sheet** modal window by parsing and retrieving the content of the sheets "on-the-fly", from an HTML file located on your site.

Enter the path/name of that file in the "HTML File" property (file from same site, without "http://").

The HTML contained in the file should follow the **"one-DIV-per-sheet" rule**, as described in the ["Embedded"](#) option (click this link to view how to structure the file!).

Contrary to the "Embedded" option, with which you enter the "one-DIV-per-sheet" HTML directly into the project, the "HTML File" option parses and retrieves the sheets' HTML "on-the-fly" from the web file.

Like the "AJAX" option of single-sheet windows, this option is useful for creating and showing database-driven or "on-the-fly" sheets, with updated content each time the modal window opens (as long as your PHP or ASP.NET, etc. refreshes this file before the modal window accesses it).

Notes:

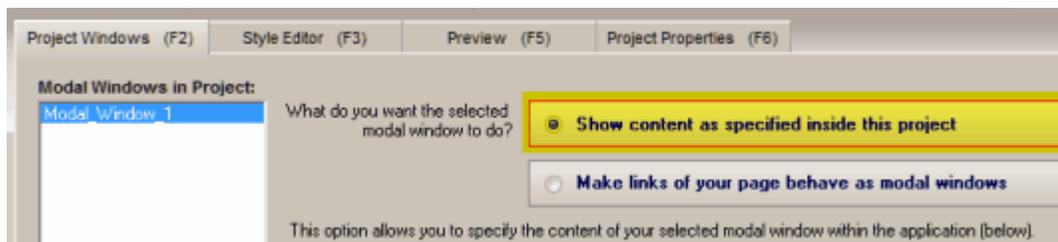
- The path of the file should be **relative** to the location of the page that opens the modal window. You **cannot** use absolute URLs, such as `http://` or `https://`
- The specified file should be present on your website, the **Preview** cannot show it!

See also

[Content options for a multi-sheet window](#)

5.3.2.4 Opens - When

Note: The following information refers to opening methods when the 1st main option is used in the "Project Windows" tab:



You have **5 methods** to make your window appear.
(non mutually exclusive, same window can be called by different elements on the same page)

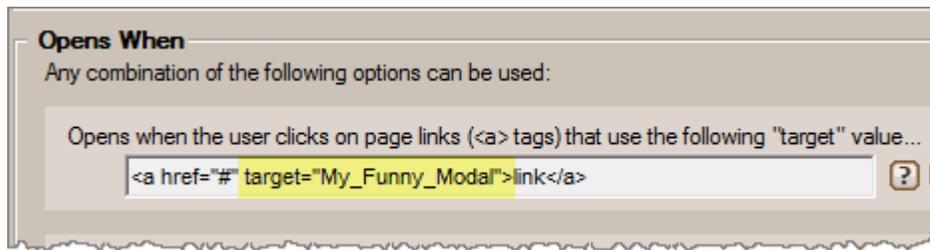
Method 1 ("Target")

Modal window can open when the user clicks on a page link (<a>) that uses the **"target"** HTML attribute with the window's name as value.

Let's assume that you name a modal window "My_Funny_Modal":



The "target" attribute should use the (case-sensitive) name of the modal window, as shown below:



Important note!

The **"href"** value is **ignored**. Why?

When the 1st main option ("*Show content as specified inside this project*") is selected, the modal window is populated by settings in the "Content" section (above the "Opens When" section), NOT by the "href" value.

Use the 2nd main option ("*Make links of your page behave as modal windows*") if you wish to use the "href" for content source.

Optional parameters: You may also add the following optional parameters to change the specified modal's width, height, header content or footer content "on demand" (i.e. on calling the modal window):

```
data-lwmw-width="400px" (or "70%", or "auto")
data-lwmw-height="300px" (or "70%", or "auto")
data-lwmw-header="your custom header content here"
data-lwmw-footer="your custom footer content here"
```

This is an example:

```
<a href="#" target="Modal_Window_1" data-lwmw-width="70%" data-lwmw-height="auto"
data-lwmw-header="This is a new HEADER!" data-lwmw-footer="This is a new FOOTER!">Info
on Product 1</a>
```

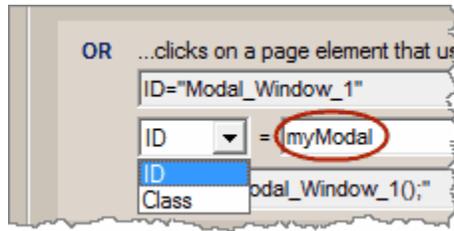
Method 2 ("ID or Class")

Modal window can open when the user clicks on a **Page Element** that uses a **at least one** of the following attributes inside its HTML code:

ID or Class

Examples:

Let's assume that you decide to enter the value **myModal** at the <Opens - When> property.



If you are using the **ID** method, add the "myModal" value to **one** element in your page, as shown below:

```
<a id="myModal" href="javascript:void(0)">Click to open Modal Window</a>
```

```
<div id="myModal">Click to open Modal Window</div>
```

```
<p id="myModal">Click to open Modal Window</p>
```

```
<span id="myModal">Click to open Modal Window</span>
```

```
<img id="myModal" href="image_here.jpg">
```

```
<table id="myModal"><tr><td>Click to open Modal Window</td></tr></table>
```

```
<input id="myModal" type="button" value="Click to open Modal Window" />
```

...or whatever other element you like.

*Note: Do NOT use the same "ID" on more than one element. The "ID" identifies a **single** element only.*

Troubleshooting on common errors

If you are using the **Class** method, add the "myModal" value to **one (or more)** elements in your page, as shown below:

```
<a class="myModal" href="javascript:void(0)">Click to open Modal Window</a>
```

```
<div class="myModal">Click to open Modal Window</div>
```

```
<p class="myModal">Click to open Modal Window</p>
```

```
<span class="myModal">Click to open Modal Window</span>
```

```
<img class="myModal" href="image_here.jpg">
```

```
<table class="myModal"><tr><td>Click to open Modal Window</td></tr></table>
```

```
<input class="myModal" type="button" value="Click to open Modal Window" />
```

...or whatever other element you like.

*Note: Contrary to the "ID" method, you **CAN** use the same "Class" on **more than one** element. In that manner you can open the same modal window by clicking on different elements of the same page. In case your elements already use another class value, HTML specification for "class" allows you to use multiple values separated by space (example: class="anotherClass **myModal**").*

Troubleshooting on common errors

Optional parameters: You may also add the following optional parameters to change the specified modal's width, height, header content or footer content "on demand" (i.e. on calling the modal window):

```
data-lwmw-width="400px" (or "70%", or "auto")
data-lwmw-height="300px" (or "70%", or "auto")
data-lwmw-header="your custom header content here"
data-lwmw-footer="your custom footer content here"
```

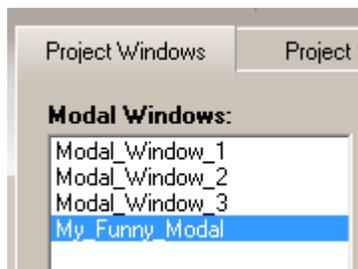
This is an example:

```
<div id="myModal" data-lwmw-width="70%" data-lwmw-height="auto" data-lwmw-header="This is a new HEADER!" data-lwmw-footer="This is a new FOOTER!">Info on Product 1</div>
```

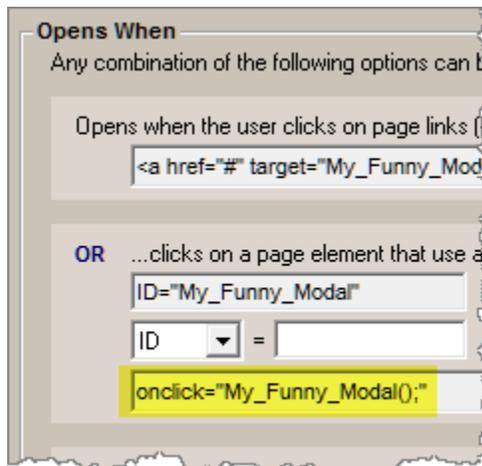
Method 3 ("onclick attribute")

Modal window can open when the user clicks on a **Page Element** that uses the "**onclick**" HTML attribute with the window's name as value.

Let's assume that you name a modal window "My_Funny_Modal":



The "onclick" attribute should use the case-sensitive name of the modal window, as shown below:



In this example, add the "myModal" value to **one (or more)** elements in your page, as shown below:

```
<a onclick="My_Funny_Modal();" href="javascript:void(0)">Click to open Modal Window</a>
```

```
<div onclick="My_Funny_Modal();">Click to open Modal Window</div>
```

```
<p onclick="My_Funny_Modal();">Click to open Modal Window</p>
```

```
<span onclick="My_Funny_Modal();">Click to open Modal Window</span>
```

```
<img onclick="My_Funny_Modal();" href="image_here.jpg">
```

```
<table onclick="My_Funny_Modal();"><tr><td>Click to open Modal Window</td></tr></table>
```

```
<input onclick="My_Funny_Modal();" type="button" value="Click to open Modal Window" />
```

...or whatever other element you like.

*Note: Contrary to the "ID" method, you CAN use the same "onclick" value on **more than one** element. In that manner you can open the same modal window by clicking on different elements of the same page.*

You can also open the modal window with a **different event**, not only with "onclick". For example you can use:

```
<a onmouseover="My_Funny_Modal();" href="javascript:void(0)">Hover mouse here to open Modal Window</a>  
(opens when you mouse over the link)
```

```
<a ondblclick="My_Funny_Modal();" href="javascript:void(0)">Double-click here to open Modal Window</a>  
(opens when you double click on the link)
```

```
<body onload="My_Funny_Modal();">...page's content...</body>  
(opens when the page loads)
```

```
<form onsubmit="My_Funny_Modal();">...form's content...</form>  
(opens when the form is submitted)
```

Optional: How to open a specific sheet in Multi-Sheet windows

If the modal window is a ["multi-sheet"](#) one, you may also use a value within the parenthesis to specify a **different sheet** to appear (other than the 1st one) when the modal window opens.

For example, to open the **3rd** sheet of a multi-sheet window you can use:

```
onclick="My_Funny_Modal(2);
```

(note that the sheet order in the API and the "onclick" function starts from **zero**, so the first sheet will be 0, the third will be 2 etc.)

You may also use a **"sheet name"** instead of a number (sheet order), if you expect that the sheet order may change in the future (by adding new sheets between the existing ones, etc.).

How?

Remember that a "multi-sheet" window is defined by consecutive DIVs ([more info](#)).

You can add an attribute called `lwmwSheetName` to the particular sheet (DIV) that you want to be displayed when the modal window opens and then use its value in the "onclick" attribute.

For example, for the following "multi-sheet" window that contains 4 sheets...

```
<div>  
  <p>This is the content of the sheet for the product  
  "iPhone"</p>  
</div>  
  
<div lwmwSheetName="product_iPad">  
  <p>This is the content of the sheet for the product  
  "iPad"</p>  
</div>  
  
<div>  
  <p>This is the content of the sheet for the product  
  "iMac"</p>  
</div>  
  
<div lwmwSheetName="product_Mac_mini">  
  <p>This is the content of the sheet for the product "Mac  
  mini"</p>  
</div>
```

...you may open the modal window with the "iPad" sheet displayed (the 2nd one), by using ANY of the 2 following attributes:

```
onclick="My_Funny_Modal(1);
```

OR

```
onclick="My_Funny_Modal("product_iPad");
```

As you realize, the 2nd implementation will still work if you later decide to change the order of the "iPad" sheet from 2nd to 1st, but requires some additional code when initially setting up the sheet's "DIV".

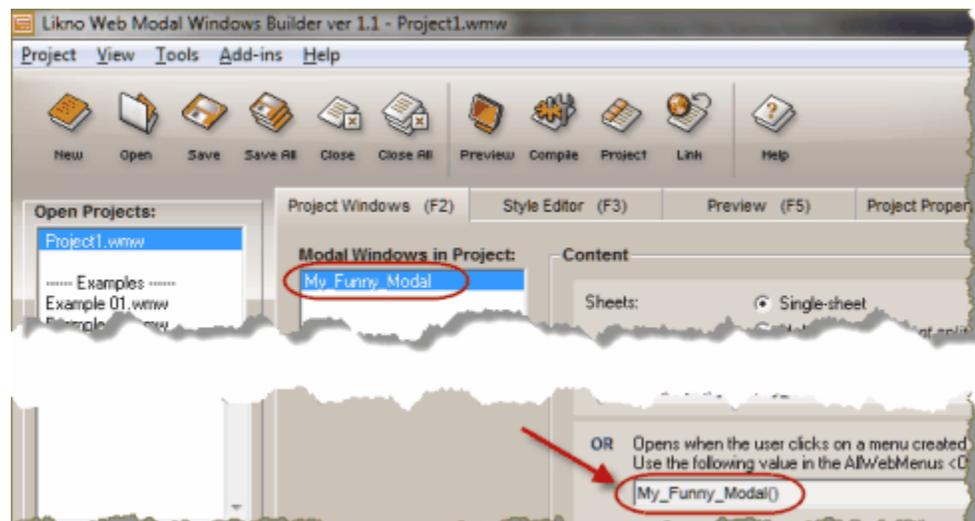
On the other hand, the 1st implementation is simpler as no additional code is required anywhere, but it is only safe as long as you are sure that the order of the sheets will remain unchanged in the future.

Method 4 (from "AllWebMenus" menu or "Likno Web Scroller Builder" scroller)

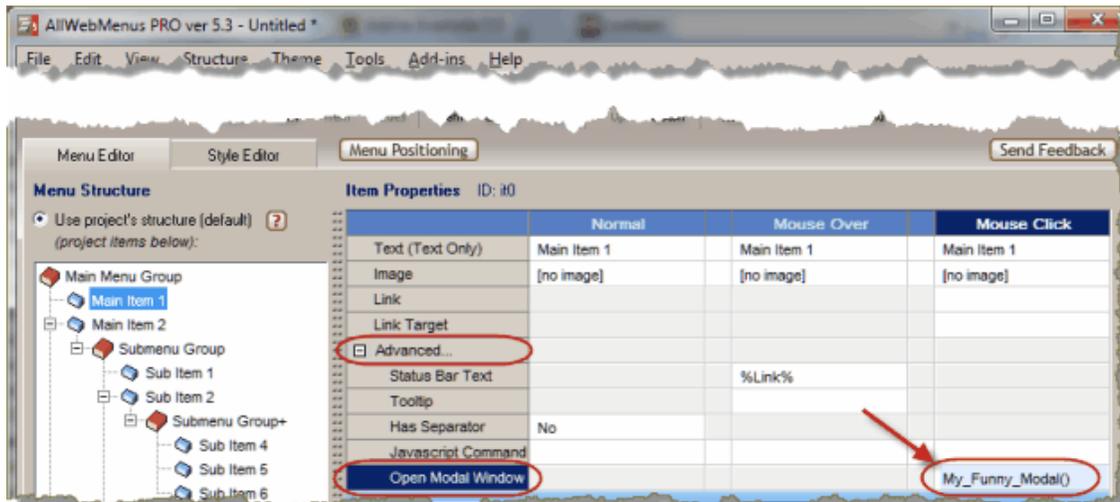
Modal window can open when the user clicks on a **menu item** of a navigation menu that you created with AllWebMenus OR a **scroller item** of a scroller that you created with Likno Web Scroller Builder.

How?

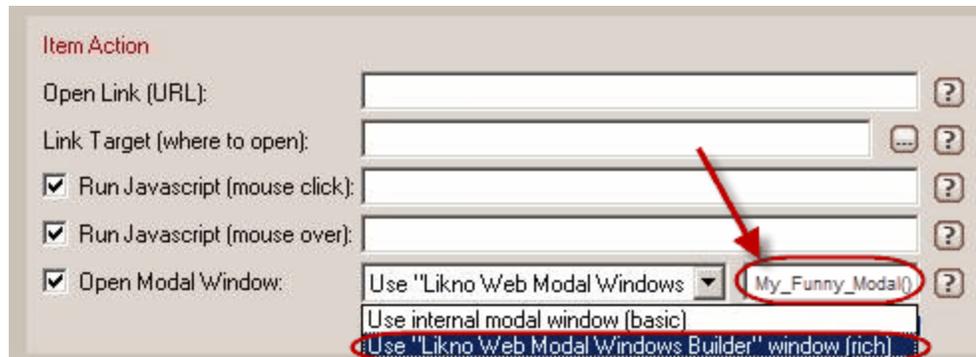
1. Copy the value shown in **Likno Web Modal Windows Builder** for the selected modal window, located here:



2. Load your menu project in **AllWebMenus** (build #792 or above) (or **Likno Web Scroller Builder** respectively).
3. In **AllWebMenus**, choose the **<Open Modal Window>** property under the "Advanced" section of your preferred **Item**. Paste the copied value there:



In the case of **Likno Web Scroller Builder**, also choose the **<Open Modal Window>** property under the "Item Action" section of your preferred scroller Item and paste the copied value there:



4. This is all. Just make sure that the compiled modal project (.js file of a .wmw project) is also linked to the web page where your menu (or scroller) appears. In pages where the modal project is not linked, "clicking" on the menu item (or scroller item) does nothing.

If the modal window is a "multi-sheet" one, you may also use a value within the parenthesis to specify a **different sheet** to appear (other than the 1st one) when the modal window opens. Read about the [proper syntax above](#) (same as with the "onclick" method).

[View more info online here](#)

Also view a [video](#) on how to use this feature!

Method 5 ("Opens on Page Load")

Modal window can open automatically when the page appears.

Just check this option and the window will automatically appear after the page is **fully loaded**. All other opening methods above still apply (for example, you may also have a link on the page that opens the same window).

Note: Only one modal window can open automatically on page load, so do not use this option on multiple windows on the same page.

You can also open a modal window on Page Load by using a [javascript variable in the page HTML or a special anchor value on its URL](#) (click to read how!).

Method 6 ("Opens on Mouse Out of Page (top side only)")

Modal window can open automatically when the user drags the mouse **outside the page**, from its top side (does not apply to the three other sides). This happens only once while the page remains open, unless the page is refreshed.

All other opening methods above still apply (for example, you may also have a link on the page that opens the same window).

Note: Only one modal window can open on Mouse Out, so do not use this option on multiple windows on the same page.

"BUT" ("Do not reopen on same user")

Modal window **can be set not to reopen** for the same user, based on some criteria:

1. "within browser session"

The modal window will not reopen for the same user until the browser is closed and opened later (**session changes**).

2. "within a given period, of X days"

The modal window will not reopen for the same user until the specified number of **X days have passed**.

3. "if user decides so"

The modal window will not reopen for the same user if the **user checks a special box** that you added in the HTML of the modal window.

The HTML code for this **special box** is formatted as follows:

```
<input type="checkbox" class="lwmw_noreopen" style="vertical-align:
-3px;"/>
```

The above code is used for the "within browser session" functionality.

If you want the "within a given period, of X days" functionality, then you add the following parameter:

```
<input type="checkbox" class="lwmw_noreopen"
lwmw_noreopen_period="10" style="vertical-align: -3px;"/>
```

...where the value "10" means "10 days".

(note: you can also use the value "session" or "0" to imply the "within browser session" functionality, but you can just remove this parameter to get the same result)

But what happens if the **user changes her mind** and wants to view the window again?

There is a way: she should delete the special cookie added by the modal window for this purpose, either manually (if she knows how) OR by clicking on a special link that you add to your page:

```
<a href="javascript:void(0);" onclick="
jQuery.cookie('projectname_windowname', null); windowname();">Force
window to reopen</a>
```

In the above code, the yellow part deletes the cookie while the purple part reopens the window (can be omitted if you want just the cookie deleted). The cookie name is composed by the **compiled project name** and the **modal window name** concatenated by an **underscore**, so make sure that use your own project's names there.

See also

Troubleshooting on common errors

[Property: Modal Window Closes - How](#)

[Property: Modal Window Closes - When](#)

[Property: Modal Window Opens - How](#)

[Modal Windows opened from Frames or IFrames](#)

5.3.2.5 Opens on Page Load

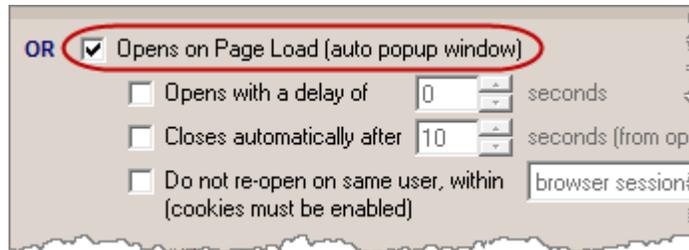
This topic elaborates further on the previous [Opens - When](#) topic, with more detail on the "Opens on Page Load (method 5)".

A modal window can open automatically when the page appears, with any of the following methods:

Method 5.1 (interface)

Specify this feature inside the project (through the interface)

Just check this option:



Note: Only one modal window can open automatically on page load (so do not use this option for multiple windows on the same page).

Method 5.2 (variable in page)

Use a javascript variable in the page that contains the modal window

In the HTML code of your page add the following code:

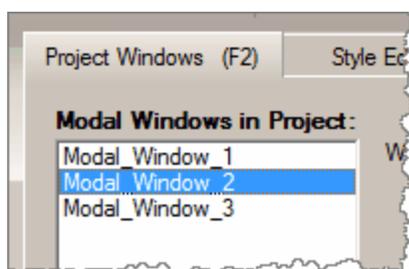
```
<script>var modalwindowname open=1;</script>
```

OR

```
<script>var modalwindowname open="SheetName";</script>
```

where:

modalwindowname is the name of your modal window here:



For example, the proper code for showing the 2nd modal window of the above project is:

```
<script>var Modal_Window_2 open=1;</script>
```

(the value "1" specifies the window's sheet if more than one exists, see below)

Notes on "Method 5.2" (variable in page):

- if the modal window is a "multi-sheet" one, you can use a different value (other than "1") in the variable to specify a **different sheet** to appear (other than the 1st one) when the modal window opens.

You may also use a "**sheet name**" instead of a number (sheet order), if you expect that the sheet order may change in the future (by adding new sheets between the existing ones, etc.).

How?

Remember that a "multi-sheet" window is defined by consecutive DIVs ([more info](#)).

You can add an attribute called `lwmwSheetName` to the particular sheet (DIV) that you want to be displayed when the modal window opens and then use its value in the variable.

For example, for the following "multi-sheet" window that contains 4 sheets...

```
<div>
  <p>This is the content of the sheet for the product
  "iPhone"</p>
</div>

<div lwmwSheetName="product_iPad">
  <p>This is the content of the sheet for the product "iPad"</p>
</div>

<div>
  <p>This is the content of the sheet for the product "iMac"</p>
</div>

<div lwmwSheetName="product_Mac_mini">
  <p>This is the content of the sheet for the product "Mac
  mini"</p>
</div>
```

...you may open the modal window "on Page Load" with the "iPad" sheet displayed, by using ANY of the 2 following HTML code in the page:

```
<script>var modalwindowname open=2;</script>
```

OR

```
<script>var modalwindowname open="product_iPad";</script>
```

As you realize, the 2nd implementation will still work if you later decide to change the order of the "iPad" sheet from 2nd to 1st, but requires some additional code when initially setting up the sheet's "DIV".

On the other hand, the 1st implementation is simpler as no additional code is required anywhere, but it is only safe as long as you are sure that the order of the sheets will remain unchanged in the future.

- if **multiple projects** use the same modal window name (in the same page), add the compiled project name in front of the variable to distinguish the window:

```
<script>var my_first_project Modal_Window_1 open=1;</script>
```

Important! If you use dashes ('-') in your compiled project name, you need to make them

underscores ('_'). For example, if your project name is `likno-modal-project.js`, the variable should be as follows:

```
<script>var likno_modal_project Modal_Window_1 open=1;</script>
```

Method 5.3 (anchor in URL)

Use a special anchor in the URL that opens the page

Open the page that contains the modal window using a special "anchor" on its URL:

```
http://www.mysite.com/mypage.html#modalwindowname_1
```

OR

```
http://www.mysite.com/mypage.html#modalwindowname_SheetName
```

For example, this is a proper URL if the name of the modal window is "Modal_Window_2":

```
http://www.mysite.com/mypage.html#Modal_Window_2_1
```

Notes on "Method 5.3" (anchor in URL):

- this method works only when the page is opened by a link from **another** page, not from the same page (because most browsers do not refresh the page when the link is from the same page).

- if the modal window is a "multi-sheet" one, you can use a different number at the end (other than "1") to specify a **different sheet** to appear (other than the 1st one) when the modal window opens.

You may also use a "sheet name" instead of a number (sheet order), if you expect that the sheet order may change in the future (by adding new sheets between the existing ones, etc.).

How?

Remember that a "multi-sheet" window is defined by consecutive DIVs ([more info](#)). You may optionally add an attribute, called `lwmwSheetName`, to the particular sheet (DIV) that you want to be displayed when the modal window opens and then use its value in the URL.

For example, for this "multi-sheet" window that contains 4 sheets...

```
<div>
  <p>This is the content of the sheet for the product
  "iPhone"</p>
</div>

<div lwmwSheetName="product_iPad">
  <p>This is the content of the sheet for the product "iPad"</p>
</div>
```

```
<div>
  <p>This is the content of the sheet for the product "iMac"</p>
</div>

<div lwmwSheetName="product_Mac_mini">
  <p>This is the content of the sheet for the product "Mac
mini"</p>
</div>
```

...you may open the modal window "on Page Load" with the "iPad" sheet displayed, by using ANY of the 2 following URLs:

```
http://www.mysite.com/mypage.html#modalwindowname_2
```

OR

```
http://www.mysite.com/mypage.html#modalwindowname_product_iPad
```

As you realize, the 2nd URL will still work if you later decide to change the order of the "iPad" sheet from 2nd to 1st, but requires some additional code when initially setting up the sheet's "DIV".

On the other hand, the 1st URL is simpler as no additional code is required anywhere, but it is only safe as long as you are sure that the order of the sheets will remain unchanged in the future.

- if **multiple projects** use the same modal window name (in the same page), add the compiled project name in front of the modal window name to distinguish the window:

```
http://www.mysite.com/mypage.html#my_first_project_Modal_Window_1_1
```

Important! If you use dashes ('-') in your compiled project name, you need to make them underscores ('_'). For example, if your project name is `likno-modal-project.js`, the variable should be as follows:

```
http://www.mysite.com/mypage.html#likno_modal_project_Modal_Window_1_1
```

- you can also add **multiple anchors** in the same URL to specify initial parameters for other Likno **controls** that exist in the same page, such as tabs, accordions, etc.!

Example:

```
http://www.mysite.com/mypage.html#my_modal_window_1#my_tab_control_3#my_accordion_c
ontrol_2
```

(for more info on their usage read the respective Help section of Likno Web Tabs Builder, Likno Web Accordion Builder, etc.)

See also

[Opens - When](#)

How Do I...

Part



6 How Do I...

6.1 Find my INSTALLATION ID

In order to activate your 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.

6.2 Purchase/Activate

Step 1. Purchase

- Access the online registration page located at <http://www.likno.com/jquery-modal-windows/purchase.php>
 - 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 to use it in your installation.

Step 2. Activate your Installation

- Choose the "**Help -> Enter Activation Code**" menu option in the application
- 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 Likno Web Modal Windows Builder 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.

6.3 Use the application with my favorite Web Authoring tool (Dreamweaver, Expression Web, etc.)

Likno Web Modal Windows Builder collaborates with all the common Web Authoring tools.

Also read: [Detailed integration info with your website \(PDF\)](#)
[FrontPage-specific integration info \(PDF\)](#)

6.4 Use an already existing element of my page as a Modal Window

It is common to open a modal window using the content of a page element that already exists in your web page. This element may as well be hidden when the page loads (for example you only have it so that it shows when the modal window opens).

You can do this through the [<Source>](#) property under the **"Project Windows"** tab. Select the "Page Element" option and identify the element by entering its ID value in the [<Page Element ID>](#) property below.

In case this element is a **form** and you wish to preserve its values between successive openings and closings of its modal window, you also need to check the *"Preserve Form Values"* option.

Note for forms: Once the modal window appears, the form will be hidden from your page and will only appear again when the modal window closes.

See also

[Property: Content Source](#)
[Page Element ID](#)

6.5 Add effects when the Modal Window opens or closes

Change the [<Modal Window Opens - How>](#) and the [<Modal Window Closes - How>](#) properties under the **"Style Editor"** tab.

See also

[Property: Modal Window Opens - How](#)
[Property: Modal Window Closes - How](#)

6.6 Change the position where the Modal Windows appear

Change the [<Modal Window Position>](#) property under the **"Style Editor"** tab.

See also

[Property: Modal Window Position](#)

6.7 Define a Header for my Modal Windows

If you want to include information in every modal window as a header, check the [<Modal Window Header>](#) property and define the content and style of the header.

Add text or HTML code in the [<Header Content>](#) property to define the content of the header. Add pure CSS code in the [<Header CSS>](#) property to define the style of the header.

See also

[Property: Modal Window Header](#)

6.8 Define a Footer for my Modal Windows

If you want to include information in every modal window as a footer, check the [<Modal Window Footer>](#) property and define the content and style of the footer.

Add text or HTML code in the **<Footer Content>** property to define the content of the footer.
Add pure CSS code in the **<Footer CSS>** property to define the style of the footer.

See also

[Modal Window Footer](#)

6.9 Load with AJAX the content of a Modal Window

You might need to load the content of one of your modal windows from another file in your server at the time you display that window. In that case you need to use the **AJAX** technique in order to load a remote HTML file and inject its contents into the DOM.

This technique is included into the application, and the only thing you need to do is to select the **"Ajax"** value in the **<Content Source>** property under the **"Project Windows"** tab, and then define the path and name of the file that you want to display in the **<Page Link>** property below.

Note! Due to security issues you are not allowed to enter an *absolute path* as the value of the call property, *nor call HTML pages that contain JavaScript code*. You can only call a file that resides within your site and contains no JavaScript code by using a *relative path*.

See also

[Content Source](#)

[Page Link](#)

6.10 Load an external site as the content of a Modal Window

You might need to load a full site, a page on a different domain, a pdf file or a page with JavaScript code as the content of one of your modal windows.

In that case you have to select the **"External Site"** value in the **<Content Source>** property under the **"Project Windows"** tab, and then define the URL of the site that you want to display in the **<Page Link>** property below.

See also

[Property: Content Source](#)

[Property: Page Link](#)

6.11 Make my modal window appear over a Flash element

Flash objects are considered to be "windows" on top of the browser and this is why they cover the modal windows 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 modal windows will always appear above Flash objects on all browsers.

6.12 Use multiple Modal Window projects in the same page

There is no restriction in the number of Modal Window projects that you can use in the same page.

You only need to define a different name for the compiled JavaScript file (our default *likno-modal-project.js*) from the Project Properties of the application and make sure to link correctly the different projects to your pages.

Note! It is very important for the modal windows of the different projects to be triggered by different elements of the page. You cannot associate an element of your page with two different modal windows.

Also read *Troubleshooting on common errors* about multiple projects and modal window names.

6.13 Link a Modal Window to an item of an AllWebMenus menu in my page?

[More info on this topic here](#)

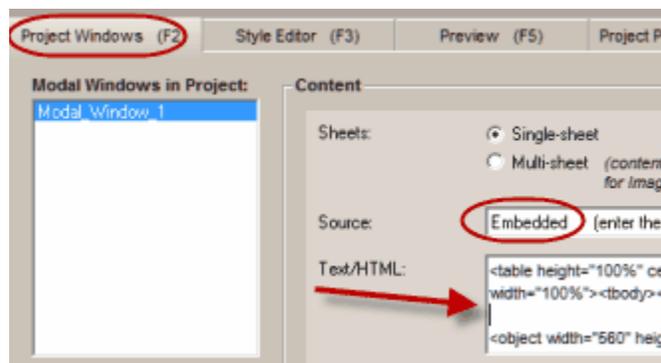
6.14 Display a video/movie in a Modal Window?

You can load and display a video/movie inside a modal window.

Open "Example 16" inside the application to see this functionality.

How is this setup?

Copy the video's "embed code" from YouTube (for example) and paste it in the "Text/HTML" property of the "Embedded" option, inside the "Project Windows" tab.



You can also change the "Window Width" and "Window Height" properties in "Style Editor" tab, to fit the width and height of the video and avoid scrollbars.

The "<table><tr><td>" section show in "**Example 16**", is used to center the video inside the Body. To show your own video, just replace the "<object>" section with its respective embed code:

```
<table height="100%" cellspacing="0" cellpadding="0" border="0"
width="100%"><tbody><tr><td align="center" valign="middle">
```

```
<object width="560" height="340">
<param name="movie"
value="http://www.youtube.com/v/d41Y9MqXrgE&hl=en&fs=1"></param>
<param name="flashvars"
value="fmt_url_map=34%7Chttp%3A%2F%2Fv12.lscache5.c.youtube.com%2Fvideo
playback"></param>
<param name="allowFullScreen" value="true"></param>
<param name="allowscriptaccess" value="always"></param>
<embed src="http://www.youtube.com/v/d41Y9MqXrgE&hl=en&fs=1"
type="application/x-shockwave-flash" allowscriptaccess="always"
allowfullscreen="true" width="560" height="340"></embed>
</object>
```

```
</td></tr></tbody></table>
```

Exactly the same would apply for other movie types, such as Flash movies.

Note: If you want to support IE6 (which is dropped by YouTube), you need to add the following parameter inside the embed code that you take from YouTube:

```
<param name="flashvars"
value="fmt_url_map=34%7Chttp%3A%2F%2Fv12.lscache5.c.youtube.com%2Fvideo
playback"></param>
```

6.15 Properly use paths in the "Website Folder" option

Contrary to the "Project Assets" option, this property is **free text** as you may need to enter both the **path** and the **filename** of the website image.

You can use **paths** to the image file in 3 different ways:

"Absolute" paths:

- <http://www.likno.com/jquery-modal-windows/assets-app/close.gif>
- <http://www.mysite.com/pics/image1.gif>
- <https://www.google.com/image2.gif>

"Document-relative" paths:

*(image folder is relative to the folder of the **loaded page** that opens the modal window)*

- close.gif
- pics/close.gif
- ../close2.gif

"Site_Root-relative" paths:

(image folder is relative to the Site Root folder)

- /close.gif
- /pics/close.gif
- /close2.gif

Note: For the Preview Panel to be able to show images using Site_Root-relative paths (example: *"/image1.gif"*) or Page-relative paths (example: *"image1.gif"* or *"pics/image1.gif"*), you need to add them to the "Project Assets" folder or to any other local folder that you specify through the **Image Folders for Preview Panel** list in [Preview Panel Settings](#).

6.16 Use the "background-image" value in CSS properties

The following is an example of CSS that applies a background image to the window through the **background-image** value:

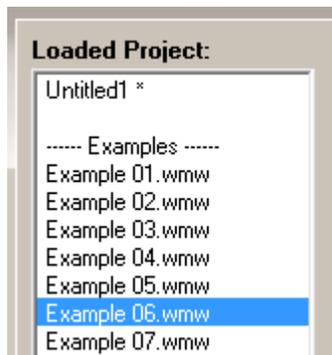
```
font-family:Verdana,Arial,Helvetica,sans-serif;font-size:11px;background-color:#e1f2c8;border:1px solid #9ecd57;padding:3px;color:#000000; background-image: url(background.jpg);
```

1. Location of the image

The project needs to know **where to find this image**, otherwise it will not be displayed in the Preview tab and will not be compiled either.

You should put this image inside the project's "Assets" folder, which is a subfolder at the folder where the project is located.

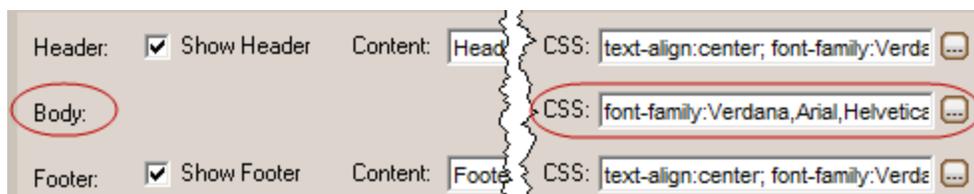
To demonstrate this, just open the "Example 06.wmw"...



... go to the "Style Editor tab...



...and edit the Body CSS value:



There you will find the CSS mentioned in the beginning, that uses the **background-image: url(background.jpg);** value.

Now, open the "assets" folder of this example, which should be at:

C:\Program Files\Likno Web Modal Windows Builder\Examples\Example 06-assets
(the installation path may be different in your computer)

There you will find the image file **background.jpg**.

If that image wasn't there, the background could not appear.

So, if you ever use a background image in your CSS, make sure that you copy this image at the project asset's folder first, otherwise the image cannot be used by the compiled project.

2. Adjusting the width/height of the window and "background-image" considerations

When a window uses a "background-image" on its CSS, this image can either **"repeat"** itself (tiles within the background) or **"no-repeat"** (shows as is, once).

This is determined by the **"background-repeat"** CSS value:

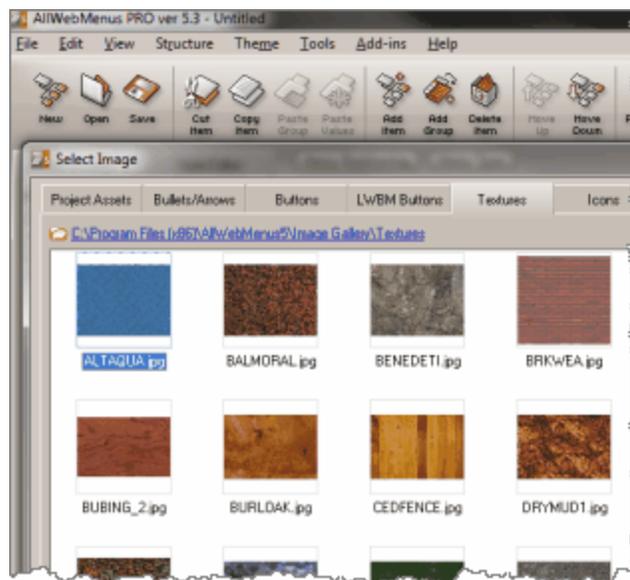
```
background-repeat: no-repeat;  
OR  
background-repeat: repeat;  
OR  
no existence of this property, which defaults to "background-repeat: repeat;"
```

Why is this important?

1. When the **"no-repeat"** value is used, the window may not show properly when you change the size ("Width" or "Height"). To show the window properly when making it bigger or smaller, you need to **also resize the background image with an image editor**.
2. This is not the case when the **"repeat"** value is used (or no value is used at all). Since the image tiles and acts as a "texture", **no image resizing is required** when changing the window size.

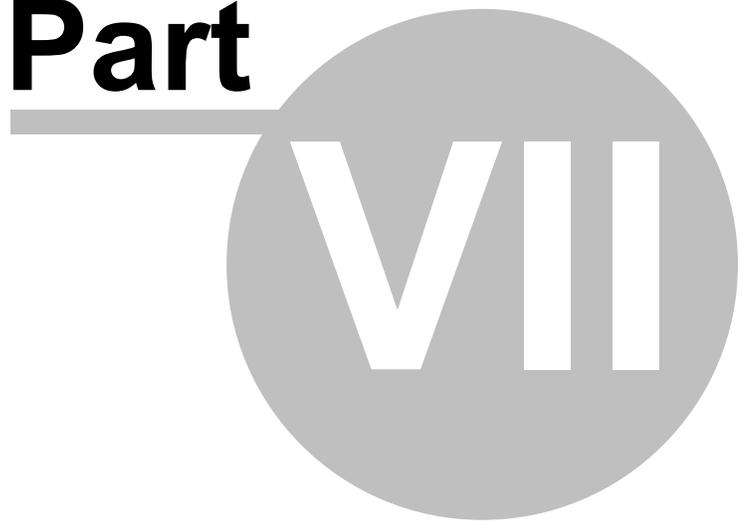
Tip on "texture" images:

[AllWebMenus](#) has plenty of textures you can copy and use as background images:



What is...

Part



7 What is...

7.1 The HTML Linking Code

This is the code added in the HTML of the page during the linking process in order to link the necessary files to your page.

This section looks like the following one (*yours may be slightly different*):

```
<!-- ***** BEGIN LIKNO WEB MODAL WINDOWS CODE FOR likno-modal *****  
-->  
<script type="text/javascript">var lwmwLinkedBy="LiknoWebModalWindows  
[1]",lwmwName="likno-modal-project",lwmwBN="100";</script><script  
charset="UTF-8" src="likno-scripts/likno-modal-project.js"  
type="text/javascript"></script></script>  
<!-- ***** END LIKNO WEB MODAL WINDOWS CODE FOR likno-modal ***** -->
```

7.2 The Header

The Header is a section that is added right on top of all the modal windows of a specific project that provides extra information.

An example of a Header is the following:



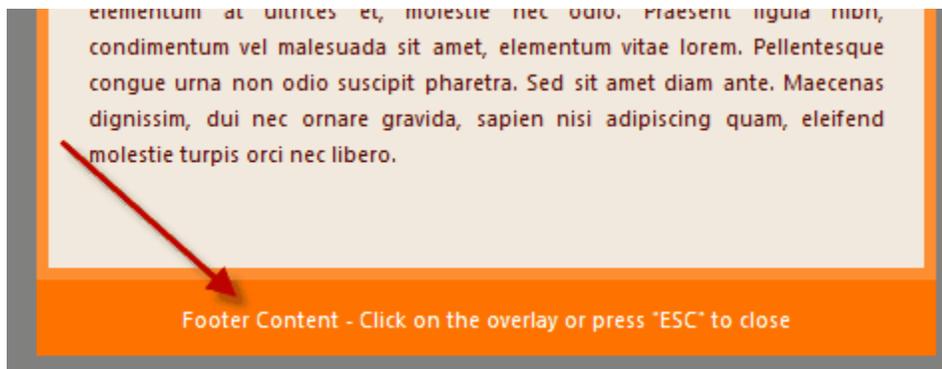
See also

[Modal Window Header](#)

7.3 The Footer

The Footer is a section that is added right at the bottom of all the modal windows of a specific project that provides extra information.

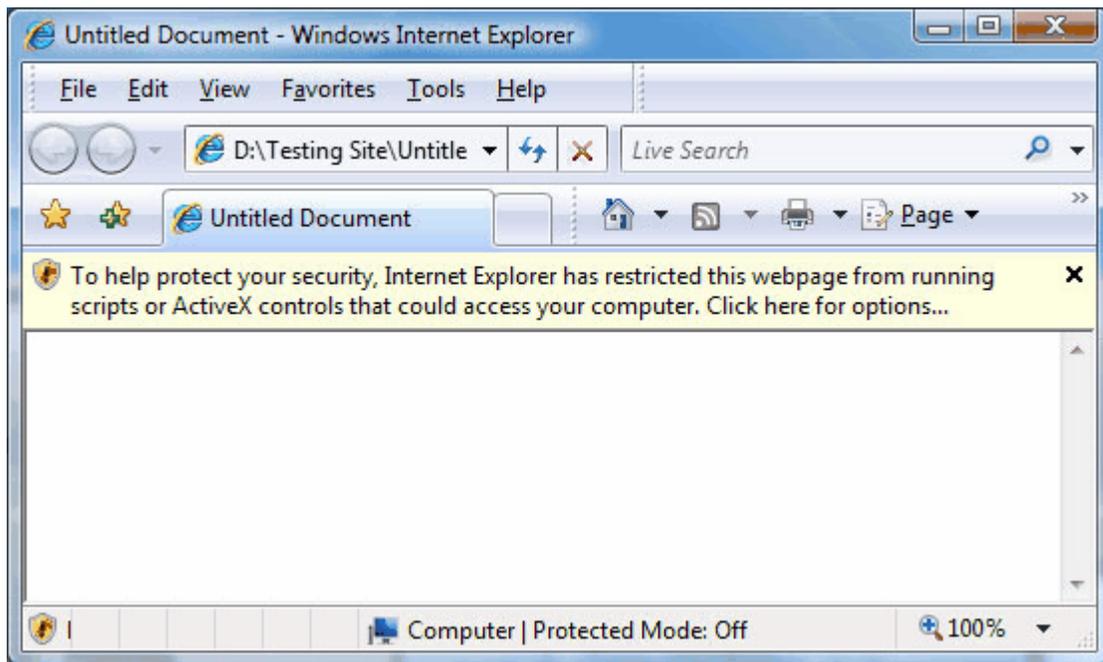
An example of a Footer is the following:

**See also**[Modal Window Footer](#)

7.4 The Internet Explorer Information Bar Warning

When you preview a **local** web page in Internet Explorer (IE6 XP SP2 +, IE7) that contains **javascript** (like our scripts 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...]

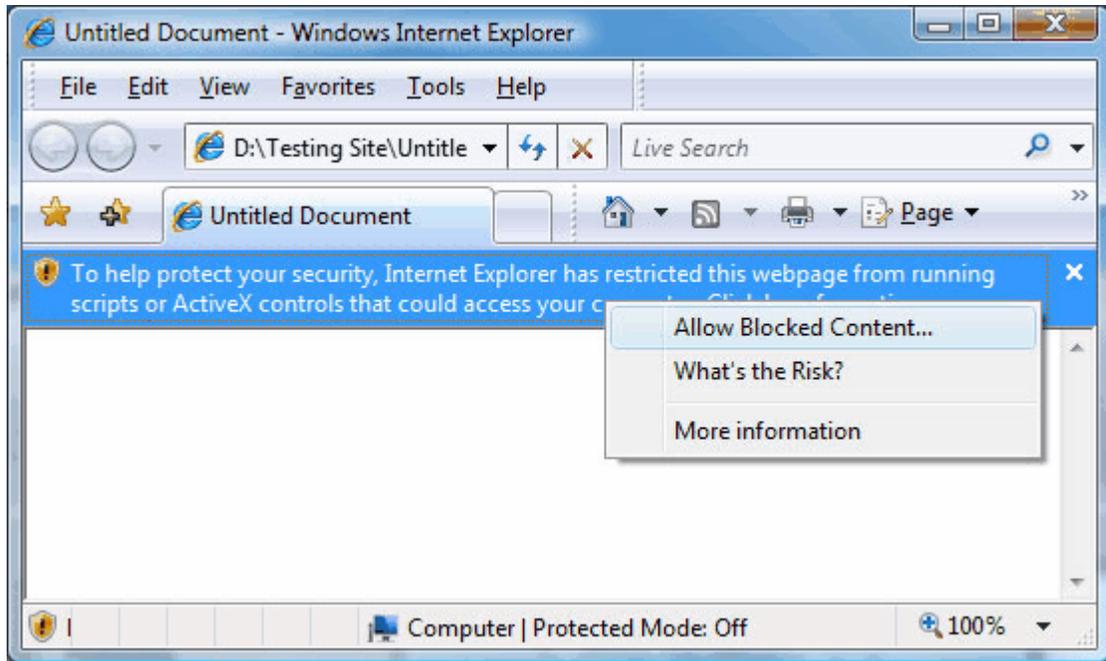


This is **not a problem**, as the "security" warning **never appears online on your actual pages**. It is not only related to Likno Software products but to all javascripts running on your pages **locally**.

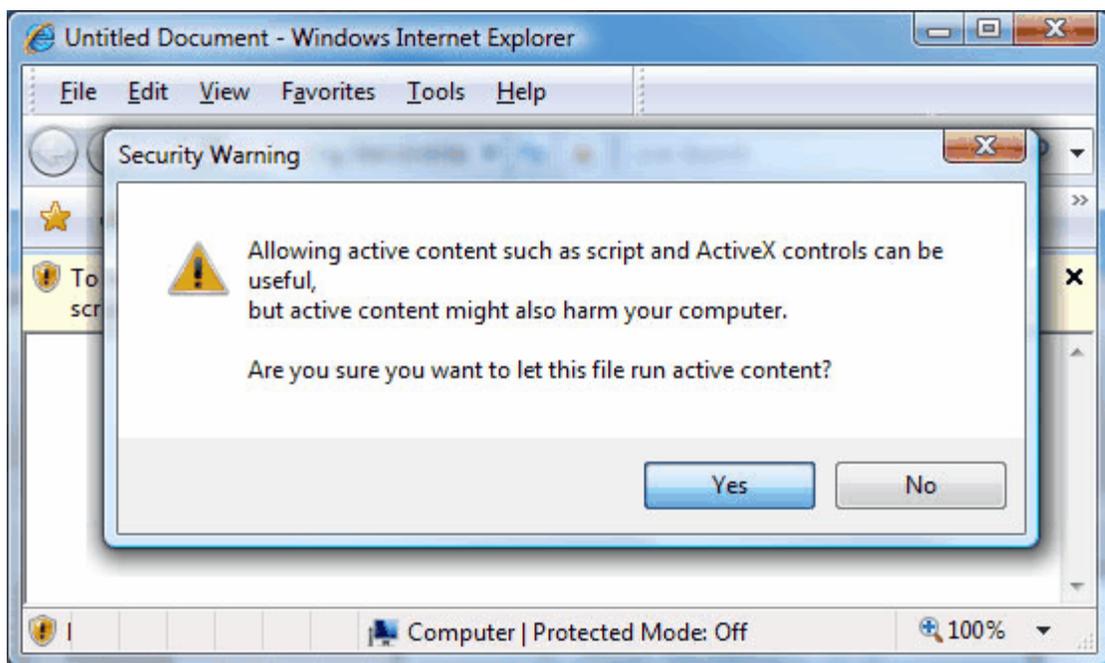
This default behavior of Internet Explorer only happens when viewing pages saved to the **local** computer and only in Internet Explorer.

This does not happen when pages have been uploaded to a web server and are then viewed online.

To overpass this Internet Explorer warning, you can click on the message bar and select **"Allow Blocked Content"**:



Then, a warning dialog will appear and you can just press **"Yes"** for the web page to properly load all the javascript related files:



(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)

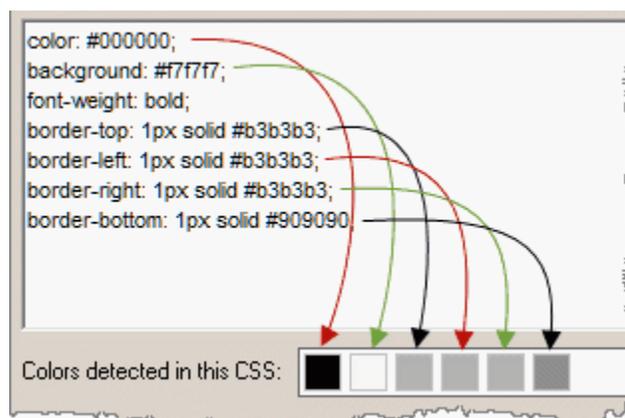
<http://windowshelp.microsoft.com/Windows/en-US/Help/91b1038d-3f53-43a3-8bd8-9551e7c0b7c61033.msp> (Windows Vista)

7.5 The "Colors detected in this CSS" feature

This feature allows you to change a color in a CSS property, by using an advanced Color Picker control instead of typing its HEX code.

How does this work?

When you edit any CSS property, the application tries to detect the colors that the CSS text is using, by detecting their HEX code:



Note that only the 3-digit or 6-digit HEX syntax (starting with a #) can be detected by the Color Picker control, such as:

```
color: #0000FF;
```

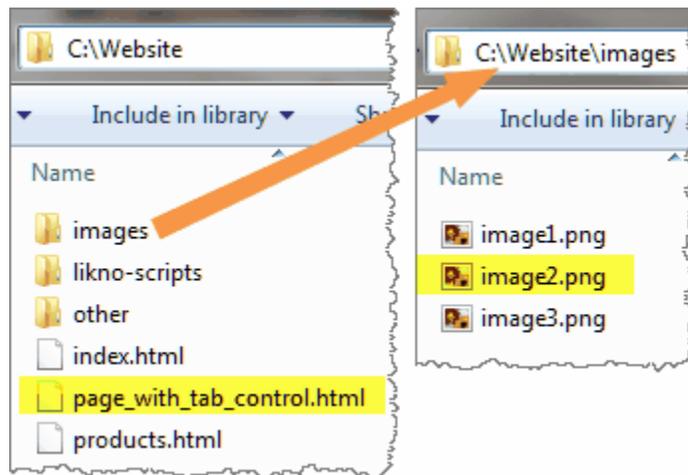
The following (valid) color syntaxes are not detected for easy editing (but are still functional as CSS values):

```
color: blue;  
color: rgb(0,0,255);  
color: rgb(0%,0%,100%);
```

7.6 The "Relative paths detected in this HTML" feature

This feature detects all "relative" (not absolute) paths used by resources (images, files, etc.) inside the HTML properties ("Embedded HTML", "Close HTML", "Header HTML", "Footer HTML", "AJAX HTML"), so that you check if these point to the correct location relative to the online page that contains the modal window.

For example, let's assume that you have the following website structure...



...and that the "page_with_tab_control.html" is the page where you have your modal window added.

If you want to show "image2.png" in a modal window, this is how its "Embedded HTML" property should look like:

```
<div style='padding: 21px 0 20px 17px; line-height: 14px;''>

Text here, next to the image...
</div>
```

OR `src="/images/image2.png"` but NOT `src="image2.png"`.

Why?

Because, the compiled modal window seeks "relative" resources (images, files, etc. that use "relative" paths in the "HTML" property) based on the page that shows it. And in this example the image is located in a subfolder of where the page is located.

As a "worry-free" alternative you can use "absolute" paths, like <http://www.likno.com/images/image2.png>.

Note for the Preview Panel only: For the Preview Panel to be able to show images using Site_Root-relative paths (example: `"/image1.gif"`) or Page-relative paths (example: `"image1.gif"` or `"pics/image1.gif"`), you need to add them to the "Project Assets" folder or to any other local folder that you specify through the **Image Folders for Preview Panel** list in [Preview Panel Settings](#).

FAQ and Tips

Part



8 FAQ and Tips

8.1 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.

I added the code in my page but nothing happens.

- Make sure that you have linked the project correctly.
- Make sure that you selected the correct domain before compiling (Project Properties).
- Make sure that you uploaded the **likno-scripts** directory after you compiled your project.

I added a second Modal Windows project to my web page and now nothing works properly.

Make sure that the two projects and their modal windows have different names.

The content of my modal window appears some pixels outside the Modal Window.

This could happen if you are using in your page the "box sizing" CSS code in order to determine the behavior or borders, paddings, etc of your elements. Our script overrides these values, however it is possible that you have used the code below:

```
* {  
    box-sizing: border-box !important;  
    -moz-box-sizing: border-box !important;  
    -webkit-box-sizing: border-box !important;  
    -ms-box-sizing: border-box !important;  
}
```

The "!important" part of the above code defines this code to override any other definition of these CSS rules. Therefore our script could not override it.

You can simply remove the "!important" part in the above definitions and your modal window will be displayed properly.

Filenames:

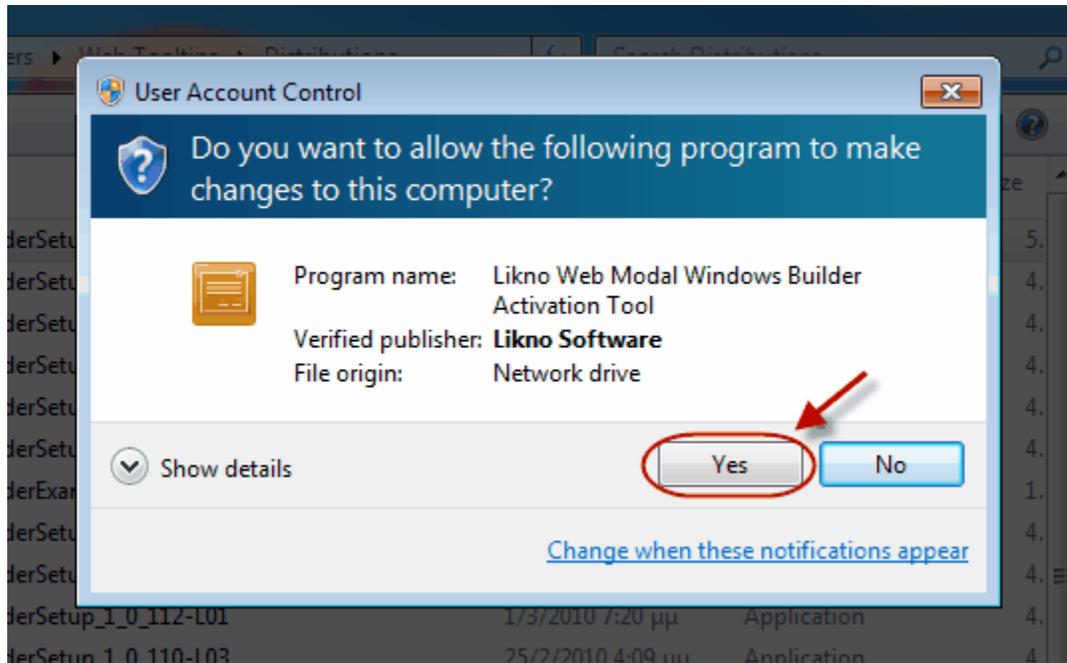
- **Spaces in the compiled project name are not allowed** as they are not parsed properly by some browsers. As an example, use "main_project.js" or "mainproject.js" instead of "main project.js".
- Avoid using filenames that contain special characters, like quotes ("), apostrophes ('), etc. In most cases these characters will produce an error.

8.2 "Full Admin rights" and the activation process

In order to activate Likno Web Modal Windows Builder you need to be an "administrator" on that PC. Otherwise, the activation process fails.

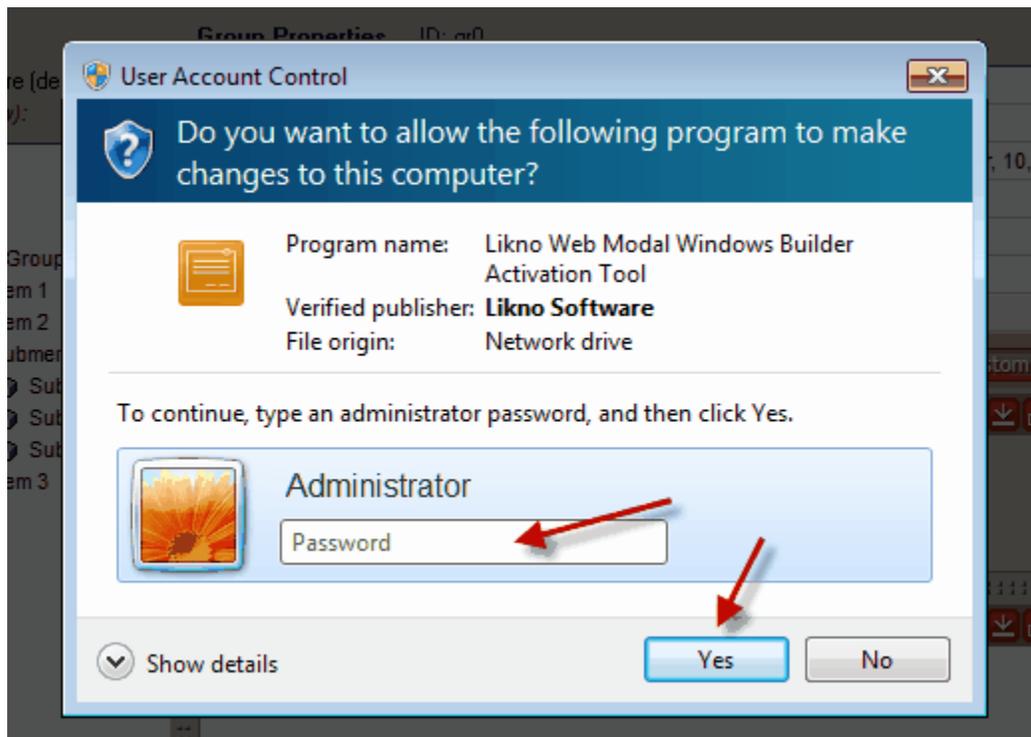
There are some cases (Windows Vista or 7 mostly) where you might think that you are administrator but you miss certain rights that are required by the activation process (you may be a "protected administrator" instead).

To get "full" administrator rights for the activation process, choose **"Yes"** on the following popup window:



(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 Likno Web Modal Windows Builder through the administrator account.



8.3 Modal Windows opened from Frames or IFrames

If you have a **frame** or **iframe** in your web page that contains links that open modal windows, please note the following:

Our engine automatically tries to **"break out"** of the frame, so that the modal windows appear centered to the browser window and not limited to the frame area only.

In some few cases and for security reasons the browsers do not allow this "break out" behavior when:

- the frame is on another domain (website) than its parent page
- the frame has a different protocol than its parent page (http vs https)
- the page is viewed locally with Google Chrome (but it does work when the same page is viewed online!)

As you can see, the above restrictions apply to very few use cases only, so in the majority of cases the behavior is as desired: the modal effect covers the entire browser window rather than the frame area only.

Just have the above notes in mind when triggering modal windows from inside a frame (or iframe).

API

Part



9 API

The **API** is used to open existing Modal Windows (already created and compiled through the application) with **different content** or **style** values than the ones you specified from within the interface (i.e. at the *"Project Windows (F2)"* and *"Style Editor (F3)"* tabs).

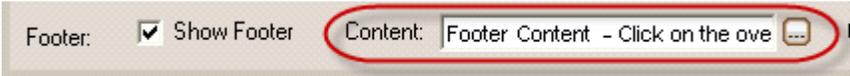
Note: In case you used customized style values through the *"Edit style of specific window"* option, then the respective customized style values will be considered as the default style values for your API calls made to that specific window.

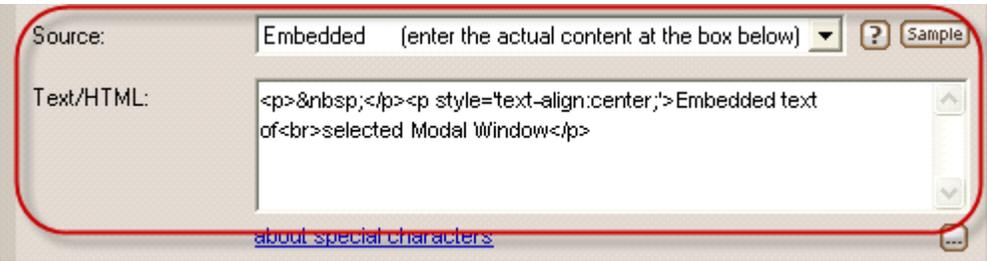
The way to call a modal window through the API is by its name, passing the parameters in the form of an object:

```
myModalName (params) ;
```

Following is a list of all the possible parameters you can pass:

header	The "header" is an object itself and holds header-specific parameters (it is not used as a parameter on its own).
show	<p>This defines whether the Header Area should be visible or not.</p> <p>For example, if you want the Header Area to be visible you should call:</p> <pre>myModalName ({header: {show: true}}) ;</pre>  <p>Values: true, false</p>
Text	<p>This defines the content of the Header Area.</p> <p>For example, if you want the Header to show "Hello world" you should call:</p> <pre>myModalName ({header: {text: 'Hello world'}}) ;</pre>  <p>Values: any text or HTML, be careful when mixing double quotes and single quotes.</p>
footer	The "footer" is an object itself and holds footer-specific parameters (it is not used as a parameter on its own).

show	<p>This defines whether the Footer Area should be visible or not.</p> <p>For example, if you want the Footer Area to be visible you should call:</p> <pre>myModalName ({ footer: { show: true } });</pre>  <p>Values: true, false</p>
text	<p>This defines the content of the Footer Area.</p> <p>For example, if you want the Header to show "Goodbye world" you should call:</p> <pre>myModalName ({ footer: { text: 'Goodbye world' } });</pre>  <p>Values: any text or HTML, be careful when mixing double quotes and single quotes.</p>
isMultiSheet	<p>This defines whether the body content should be presented as a "multi-sheet" or a "single-sheet" modal window.</p> <p>For example, if you compiled your project as "Multi-sheet" and you rather want to open this modal window as "Single-sheet" for a specific body content, then you should call:</p> <pre>myModalName ({ isMultiSheet: false });</pre>  <p>Notes:</p> <ul style="list-style-type: none"> - If you have a "mixed" case where you want the same modal window sometimes to open as "Single-Sheet" and sometimes as "Multi-Sheet", then you should make sure that you compile it as "Multi-Sheet", otherwise the related graphics (navigation bar, etc.) will not be present when needed ("Single-Sheet" windows do not compile these by default). This is actually the case where the <code>{isMultiSheet: false}</code> parameter is useful, since you can open single-sheet modal windows even from a multi-sheet compiled project. - If you compile your modal window as "Single-Sheet" you do not need to use the <code>{isMultiSheet: false}</code> parameter, it is considered to be as such by default.

	<p>- If you compile your modal window as "Multi-Sheet" you do not need to use the <code>{isMultiSheet: true}</code> parameter, it is considered to be as such by default.</p> <p>Values: true, false</p>
currentSheet	<p>This defines which sheet will display as selected once the modal window opens.</p> <p>Note! please keep in mind that the numbering starts from 0, so if you want the third sheet to be displayed you should use:</p> <pre>myModalName ({currentSheet:2});</pre>
bodySource	<p>This defines the content of the Body Area.</p>  <p>Depending on the value you pass and the value of the "isMultiSheet" parameter you may get different results.</p> <p>If you have selected a single-sheet modal window the value can be:</p> <ul style="list-style-type: none"> - Some HTML, in which case the HTML will be shown as is. - A page element (through a jQuery selector). - A single <code></code> tag, in which case it will be treated as an image-only window, and the image will be centered within the body area. - A single <code><div></code> tag with a property "ajaxUrl", in which case the content of the given URL will be loaded (by Ajax) as the content of the body area. - A single <code><div></code> tag with a property "externalUrl", in which case the content of the given URL will be loaded (inside an iFrame) as the content of the body area. <p>Here are some single-sheet examples you can try:</p> <pre>myModalName ({bodySource:'Hello world'}); myModalName ({bodySource:lQuery('#myElement')}); myModalName ({bodySource:'<div>Hello <u>world</u></div>'}); myModalName ({bodySource:''});</pre>

	<pre>myModalName ({bodySource: '<div ajaxUrl=\'myAjaxFile.html\' />'}); myModalName ({bodySource: '<div externalUrl=\'http://www.google.com/\' />'});</pre> <p>If on the other hand you have selected a multi-sheet modal window, the value must be a series of <div> tags each containing the content of one sheet. The same rules apply for each sheet (each <div>), as for the single-sheet content.</p> <p>Here are some multi-sheet examples you can try:</p> <pre>myModalName ({bodySource: '<div>Hello</div><div>world</div><div>!</div>'}); myModalName ({bodySource: '<div></div><div>Hi</div><div ajaxUrl=\'mypage.html\' /></div>'}); myModalName ({bodySource: jQuery('.myClass')});</pre> <p>(in the last example we select to display all page elements with class "myClass", each in a separate sheet)</p> <p>Notes:</p> <ul style="list-style-type: none"> - In the ID and CLASS examples you read it right, it is "jQuery" and not "jQuery". - Be careful when mixing double quotes and single quotes and make sure that you escape these characters when used inside the parameter.
containerCss	<p>This defines the basic formatting of the modal window and you can pass most common CSS values (at your own risk though, as some combinations may "break" the modal window).</p> <p>The values should be passed as an object, with the attributes separated by commas and each attribute name and value within quotes. This is very important, as having a semi-colon instead of comma or omitting the quotes may generate JavaScript errors.</p> <p>For example if you want to open your existing modal window as 400 pixels wide and 300 pixels tall with a solid 3-pixel red border, you should call:</p> <pre>myModalName ({containerCss: { 'width': '400px', 'height': '300px', 'border': 'solid #FF0000 3px'}});</pre> 

You can use **several parameters** in a single call, just **separate them with spaces**.

Here is an example of how you can pass several parameters at once (the parameter order does not matter):

```
myModalName({containerCss: {'width':'400px', 'height':'300px'},  
header:{show:true, text:'a nice header'}, bodySource:'hello there',  
isMultiSheet:false, footer:{show:false}});
```

Index

- A -

About Likno Software 4
Activate 71
Activation 86
Add to page 6
Admin 86
Administrator 86
Adobe Golive 71
AJAX 34, 51, 73
AJAX rules 51
API 90
Applet 73
Asynchronous 34

- B -

Bottom 23

- C -

Click 21, 22, 56
Close Selected Project 12
Closes how 21
Closes when 22
Closing 21, 22
CoffeeCup HTML Editor 71
Compile Project 12
Compile Properties 14
Content 6, 37, 47
Content source 48, 49, 50, 51, 52, 54, 55
Cool Page 71
Create Web Modal Windows 6
CSS 23, 26, 27
CuteHTML 71

- D -

Dark area 31
Darkened area 31
DreamWeaver 71
Dynamic 73

- E -

Edit 7
Effect 20, 21
Effects 6
Element 50, 54
Embedded 49, 52
Exit 7, 12
Expression Web 71
External 51

- F -

Flash 73, 74
Footer 33
Form 37, 46
Frame 88
FrontPage 71
Ftp 7

- G -

Generate files 6
Group 37, 47

- H -

Header 31
Height 25
Help 18
HomeSite 71
HotDog 71
How Do I
 Add effects when the Modal Window opens or
 closes 72
 Change the position where the Modal Windows
 appear 72
 Define a Footer for my Modal Winodws 72
 Define a Header for my Modal Winodws 72
 Display a video/movie in a Modal Window? 74
 Find my INSTALLATION ID 71
 Link a Modal Window to an item of an
 AllWebMenus menu in my page 74
 Load an external site as the content of a Modal
 Window 73

How Do I

Load with AJAX the content of a Modal Window
73

Make my modal window appear over a Flash
element 73

Purchase/Activate 71

Use an already existing element of my page as a
Modal Window 72

Use Likno Web Modal Window Builder with my
favorite Web Authoring tool 71

Use the "background-image" value in CSS
properties 76

Use the "Web Folder" paths 75

Use two different Modal Window projects in the
same page 74

HTML File 55

HTML Kit 71

- I -

iFrame 88

Image 50

Installation ID 71

- K -

Keyboard shortcuts 9

- L -

Layer 27

Layout 9

Left 23

Library 6

Likno Web Modal Windows Builder information 2

liknomodal-close 22

Link 6

Link Compiled Project to Web Pages 13

- M -

Main window layout 9

Multiple Sheets 49, 52, 54, 55

Multi 37, 47

Multi-Sheet 48

- N -

Navigation Bar 34

NetObjects Fusion 71

New 12

- O -

Object 73

Opacity 31

Open 12

Opening 20, 21

Opens how 20

Opens on Page Load 65

Opens when 21, 56, 65

Options 6

Overlay 31

- P -

Page links 37, 47

Position 23

Preview Panel Settings 16

Project 12

Project Properties 14

Properties Overview 20

Publish 7

Purchase 71

- R -

Register 71

Right 23

Rounded Corners 27

- S -

Save 7, 12

Save As 12

Settings 6

Shadow 30

Sheets 48

Shortcuts 9

Single 37, 47

Single Sheet 49, 50, 51

Single-Sheet 48
SlideShow 34
Style 23, 25, 26, 27, 30, 31

- T -

Target attribute 37, 47
Terminology 3
Top 23
Transparency 31
Trigger 21, 22, 56
Troubleshooting 86

- U -

UAC 86
Under the Hood 17
Upload 7

- V -

Video 74
Visual InterDev 71

- W -

Web Authoring tool 71
Web Server 73
What is
 The "Colors detected in this CSS" feature 83
 The "Relative paths detected in this HTML"
 feature 83
 The Footer 80
 The Header 80
 The HTML menu-linking code 80
 The Internet Explorer Information Bar Warning
 81
What's new! 3
Width 25

- Y -

YouTube video 74

- Z -

Z-index 27



<http://www.likno.com>