Likno Web Modal Windows Builder

API Help Document

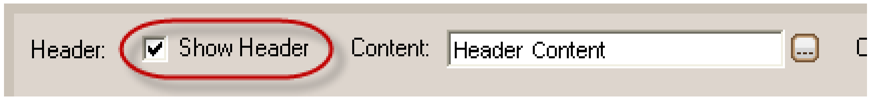
The **API** is used to open existing Modal Windows (already created and compiled through the application) with **different content** or **style** values than the ones you specified from within the interface (i.e. at the *"Project Windows (F2)"* and "*Style Editor (F3)"* tabs).

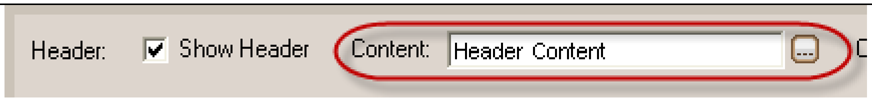
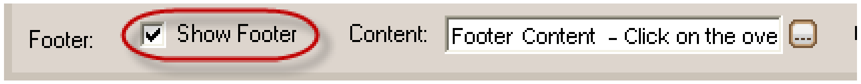
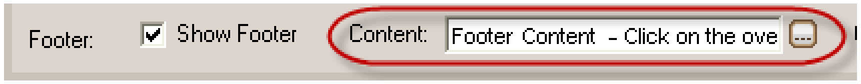
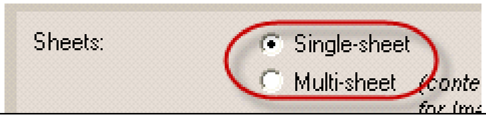
Note: In case you used customized style values through the *"Edit style of specific window"* option, then the respective customized style values will be considered as the default style values for your API calls made to that specific window.

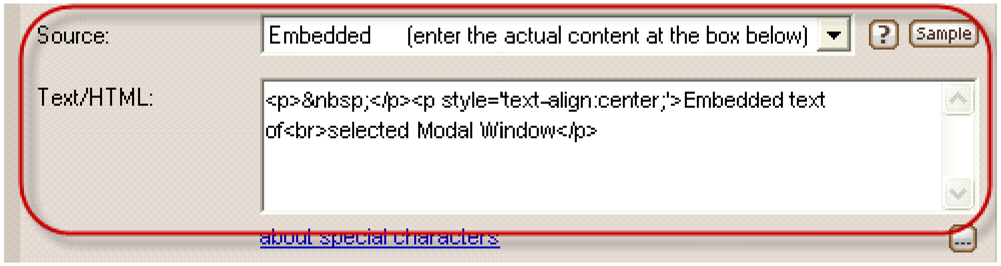
The way to call a modal window through the API is by its name, passing the parameters in the form of an object:

```
myModalName(params);
```

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

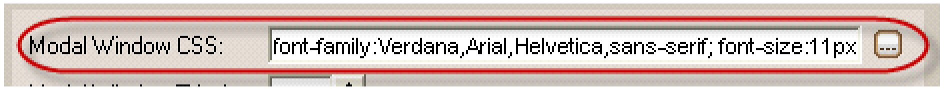
`header`	The "header" is an object itself and holds header-specific parameters (it is not used as a parameter on its own).
`show`	This defines whether the Header Area should be visible or not. For example, if you want the Header Area to be visible you should call: `myModalName({header:{show:true}});`  Header: ☑ Show Header Content: Header Content Values: true, false
`Text`	This defines the content of the Header Area. For example, if you want the Header to show "Hello world" you should call: `myModalName({header:{text:'Hello world'}});`

	 Header: ☑ Show Header Content: Header Content Values: any text or HTML, be careful when mixing double quotes and single quotes.
`footer`	The "footer" is an object itself and holds footer-specific parameters (it is not used as a parameter on its own).
`show`	This defines whether the Footer Area should be visible or not. For example, if you want the Footer Area to be visible you should call: `myModalName({footer:{show:true}});`  Footer: ☑ Show Footer Content: Footer Content - Click on the ove Values: true, false
`text`	This defines the content of the Footer Area. For example, if you want the Footer to show "Goodbye world" you should call: `myModalName({footer:{text:'Goodbye world'}});`  Footer: ☑ Show Footer Content: Footer Content - Click on the ove Values: any text or HTML, be careful when mixing double quotes and single quotes.
`isMultiSheet`	This defines whether the body content should be presented as a "multi-sheet" or a "single-sheet" modal window. 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: `myModalName({isMultiSheet:false});`  Sheets: ● Single-sheet ○ Multi-sheet (conte for (ma

	Notes: - 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 `{isMultiSheet:false}` 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 `{isMultiSheet:`**`false`**`}` parameter, it is considered to be as such by default. - If you compile your modal window as "Multi-Sheet" you do not need to use the `{isMultiSheet:`**`true`**`}` parameter, it is considered to be as such by default. Values: true, false
`currentSheet`	This defines which sheet will display as selected once the modal window opens. Note! please keep in mind that the numbering starts from 0, so if you want the third sheet to be displayed you should use: `myModalName({currentSheet:2});`
`bodySource`	This defines the content of the Body Area.  Source: Embedded (enter the actual content at the box below) Text/HTML: <p>&nbsp;</p><p style="text-align:center;">Embedded text of selected Modal Window</p> about special characters Depending on the value you pass and the value of the "`isMultiSheet`" parameter you may get different results. If you have selected a single-sheet modal window the value can be: - Some HTML, in which case the HTML will be shown as is. - A page element (through a jQuery selector). - A single <img> 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 <div> 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 <div> 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.
	Here are some single-sheet examples you can try:
	`myModalName({bodySource:'Hello world'});`
	`myModalName({bodySource:lQuery('#myElement')});`
	`myModalName({bodySource:'<div><b>Hello</b> <u>world</u></div>'});`
	`myModalName({bodySource:'<img src=\'myImage.jpg\'/>'});`
	`myModalName({bodySource:'<div ajaxUrl=\'myAjaxFile.html\'/>'});`
	`myModalName({bodySource:'<div externalUrl=\'http://www.google.com/\'/>'});`
	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.
	Here are some multi-sheet examples you can try:
	`myModalName({bodySource:'<div>Hello</div><div>world</div><div>!</div>'});`
	`myModalName({bodySource:'<div><img src=\'myImage.jpg\'/></div><div>Hi</div><div ajaxUrl=\'mypage.html\'></div>'});`
	`myModalName({bodySource:lQuery( '.myClass')});`
	(in the last example we select to display all page elements with class "myClass", each in a separate sheet)
	Notes:
	- In the ID and CLASS examples you read it right, it is "lQuery" 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`	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). 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. 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:

	myModalName({containerCss:{'width':'400px','height':'300px','border':'solid #FF0000 3px'}});  Modal Window CSS: font-family:Verdana,Arial,Helvetica,sans-serif; font-size:11px

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}});
```