



Creating your first Adobe® GoLive® CS 2 Extension

**A step by step guide to create your first
extension in GoLive CS2.**

The GoLive Extend Script Software Development Kit (SDK) is a powerful tool that allows you to extend the functionality of GoLive.

Using the Extend Script SDK, you can create an extension that:

- Adds menus, dialogs, and palettes dynamically
- Custom property inspectors
- Drag-and-drop custom HTML elements that can be edited in the inspector
- Access remote data using the FTP or WebDAV objects

This step by step guide will walk you through the steps to create your first extension. A knowledge of JavaScript and XML markup is helpful but not required. After completing this guide, you will have:

- Installed tools for creating extensions
- Created an extension that adds a menu item and display a dialog.

1.0 Before you begin

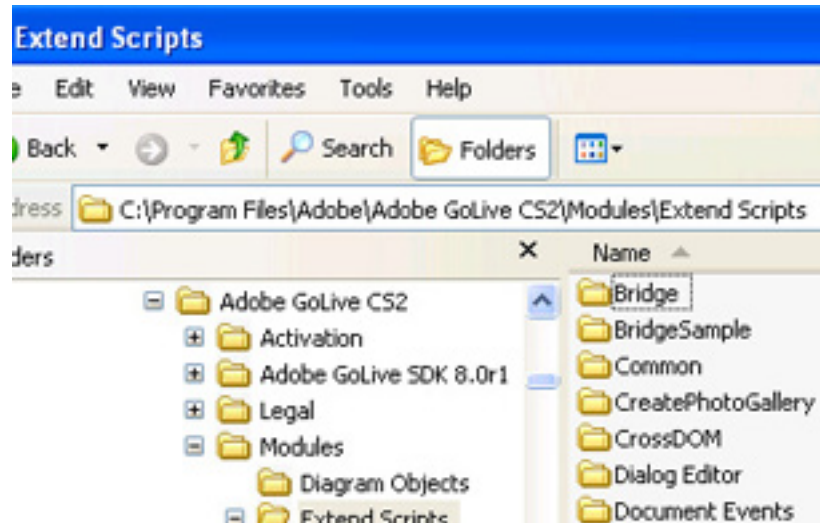
Before you begin, you will want to make sure of the following.

1.1 GoLive installation

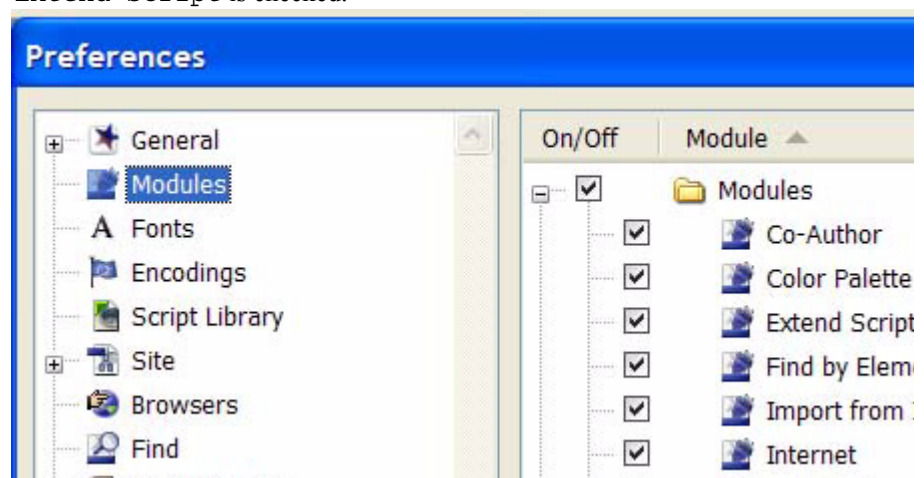
1. Make sure that GoLive CS2 is installed.
2. Copy the following folders from the Adobe GoLive SDK 8.0r1/Tools folder to the GoLive/Modules/Extend Scripts folder:
 - Dialog Editor

Before you begin

- Edit Extension
- Extension Builder

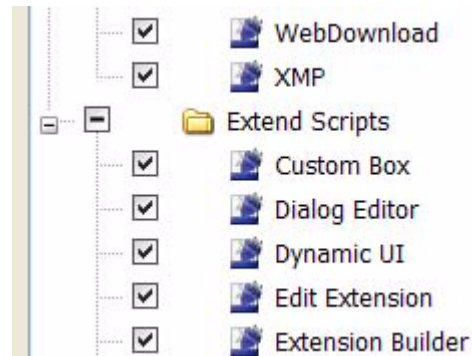


3. Start GoLive.
4. Make sure the Extend Scripts module and 3 extensions from step 2 are turned on.
Select Edit > Preferences > Modules, make sure that Modules > Extend Script is checked.

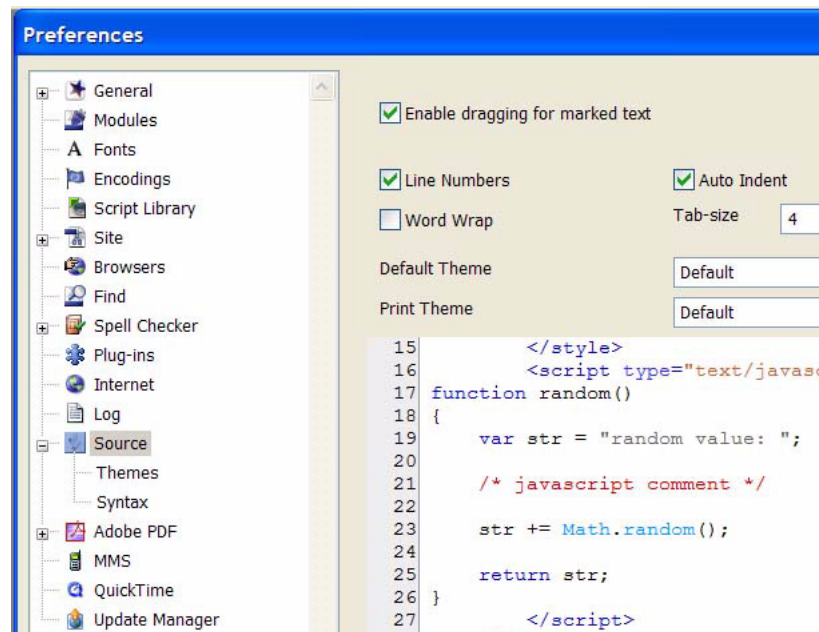


Before you begin

In the same dialog, scroll down to Modules > Extend Scripts. Dialog Editor, Edit Extension, and Extension Builder should be listed and checked.



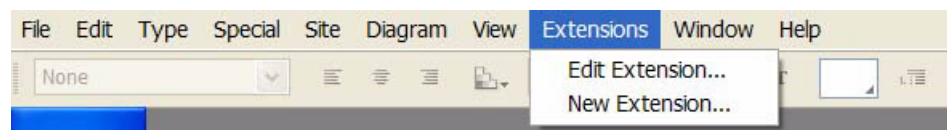
5. For easier JavaScript editing, select Edit > Preferences. Select Source and Auto Indent.



6. If you made any changes, restart GoLive for changes to take effect.

1.2 Checking the installation

If the steps in [Section 1.1: GoLive installation](#) were completed, there should be an Extensions menu with two menu items, Edit Extension and New Extension.



1. No Extensions menu
Folders in Tools not copied to correct directory (step 2).
The Extend Script module is not turned on (step 4).
Extensions disabled (step 4).
2. Extensions > Edit Extension missing
Edit Extension folder not copied to correct directory (step 2).
Edit Extension sample is disabled (step 4).
3. Extensions > New Extension missing
Extension Builder folder not copied to correct directory (step 2).
Extension Builder is disabled (step 4).
4. Window > Objects palette should have a Dialog Editor tab.
Dialog Editor folder not copied to correct directory (step 2).
Dialog Editor is disabled (step 4).

2.0 Creating the extension

Now that the installation is complete, you're ready to begin creating your first extension.

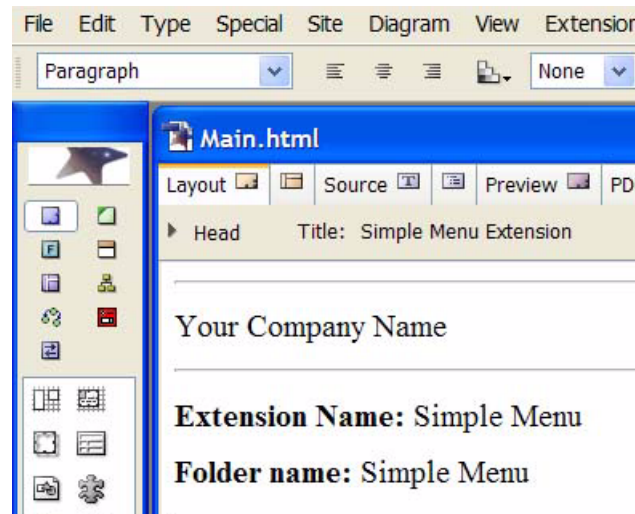
1. From the Extensions menu, select New Extension... This will bring up a dialog titled Create new Extend Script extension.



2. Fill in Simple Menu for the Extension Name.
3. Make sure that Save in GoLive/Modules/Extend Scripts folder is selected.
4. Click OK.
A folder named Simple Menu is created in GoLive/Modules/Extend Scripts.

Editing the extension

A file called `Main.html` is created in this folder, and is opened in GoLive. This is the main file for developing the extension.



3.0 Editing the extension

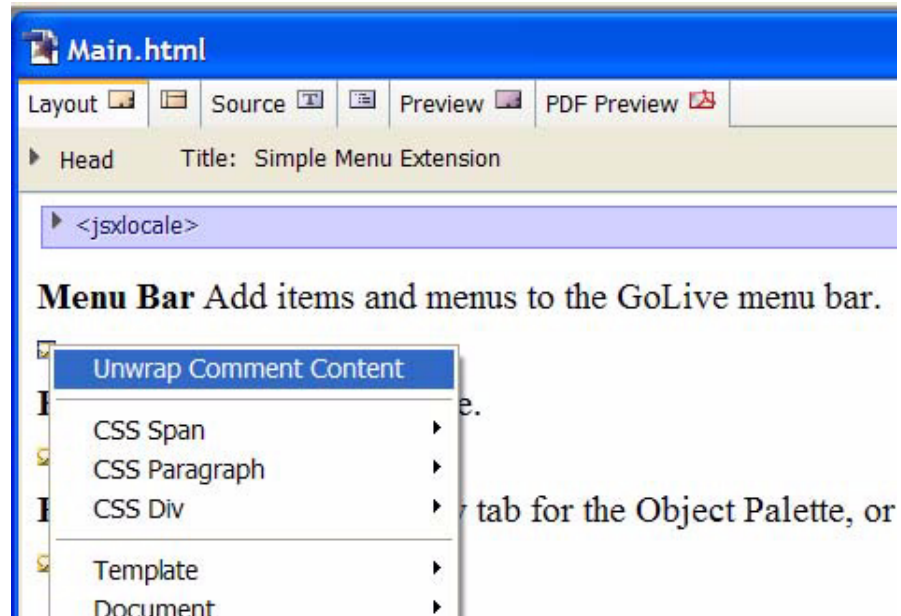
Now you're ready to begin adding functionality to the Simple Menu extension.

3.1 Adding the menu item

Follow these steps to create tags for the Simple Menu item.

1. In `Main.html` for Simple Menu, scroll down until you see the bold words **Menu Bar**.
2. Select the comment below Menu bar, and with the right mouse button, select **Unwrap Comment Content** from the pulldown menu. If you are on Macintosh, select the

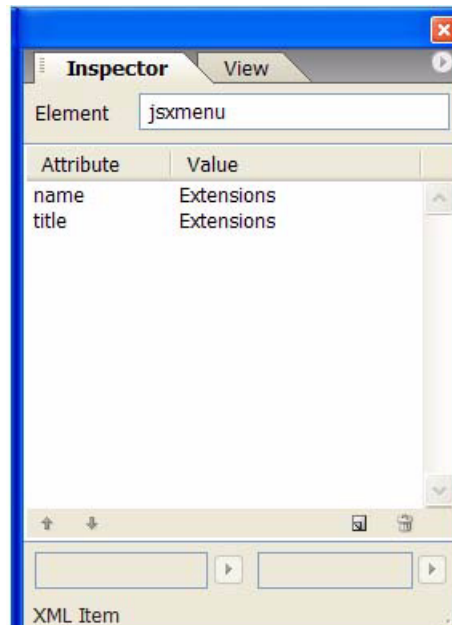
comment, and control click to get the pull-down menu. When the comment is unwrapped, the `<jsxmenubar>` tag appears.



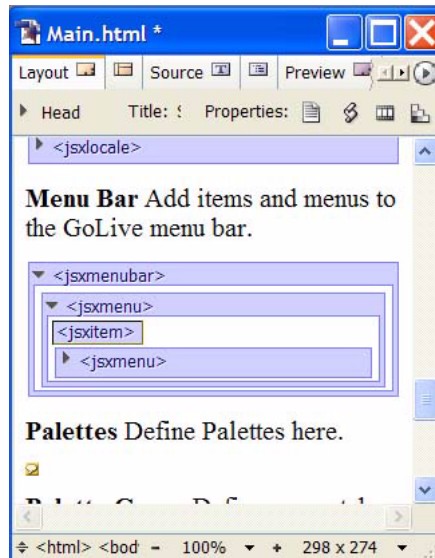
3. Click on the right pointing arrow in `<jsxmenubar>` to expand the contents of this tag.
4. Expand the enclosed `<jsxmenu>` tag.
5. Select the first `<jsxmenu>` tag.



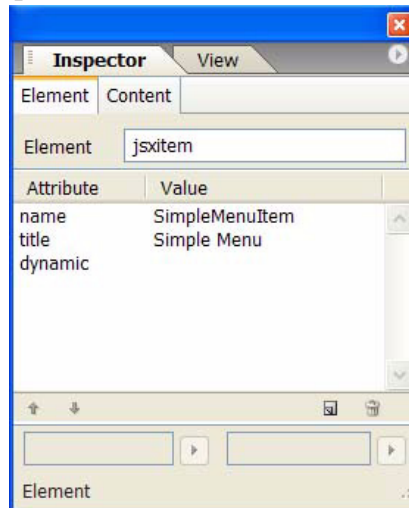
6. In the Inspector, set the attribute Name to Extensions, and set the attribute Title to Extensions.



7. Select the first <jsxitem>.



8. In the Inspector, set the attribute Name to SimpleMenuItem, and Title to Simple Menu.

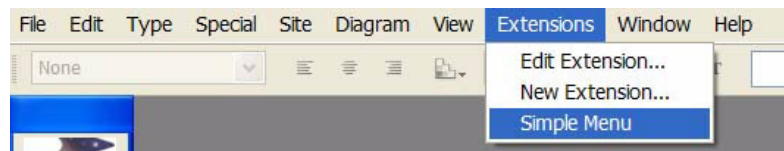


9. We will not be using the enclosed `<jsxmenu>` tag, so select and delete the enclosed `<jsxmenu>` tag.
10. Save Main.html.

3.2 Viewing the Simple Menu item

To see the Simple Menu item you'll need to restart GoLive.

Select Extensions > Simple Menu Item.



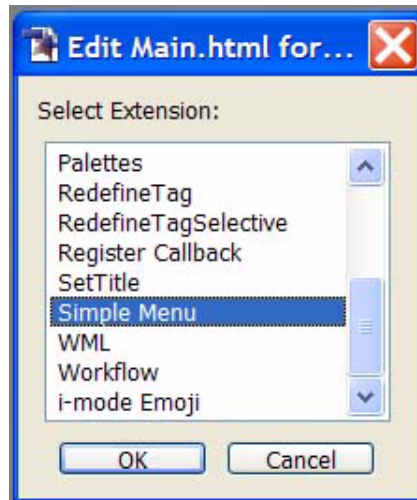
Nothing happens yet! This is because you haven't attached functionality to the menu item yet.

3.3 Adding the menu code

Now that the menu item code is there, let's attach functionality to the menu item.

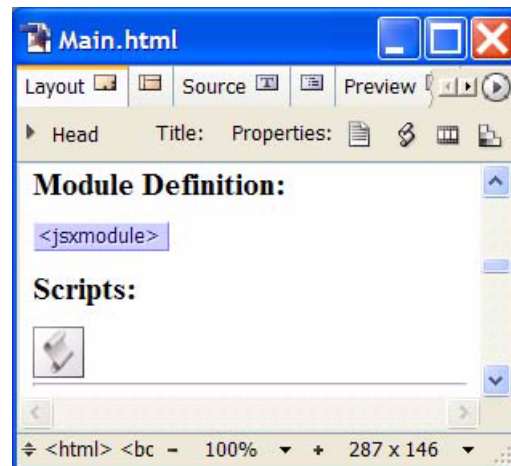
1. From the Extensions menu, select Edit Extension....

2. The Edit Main.html for Extension dialog appears. Select **Simple Menu** and click OK.



Main.html for Simple Menu opens in GoLive.

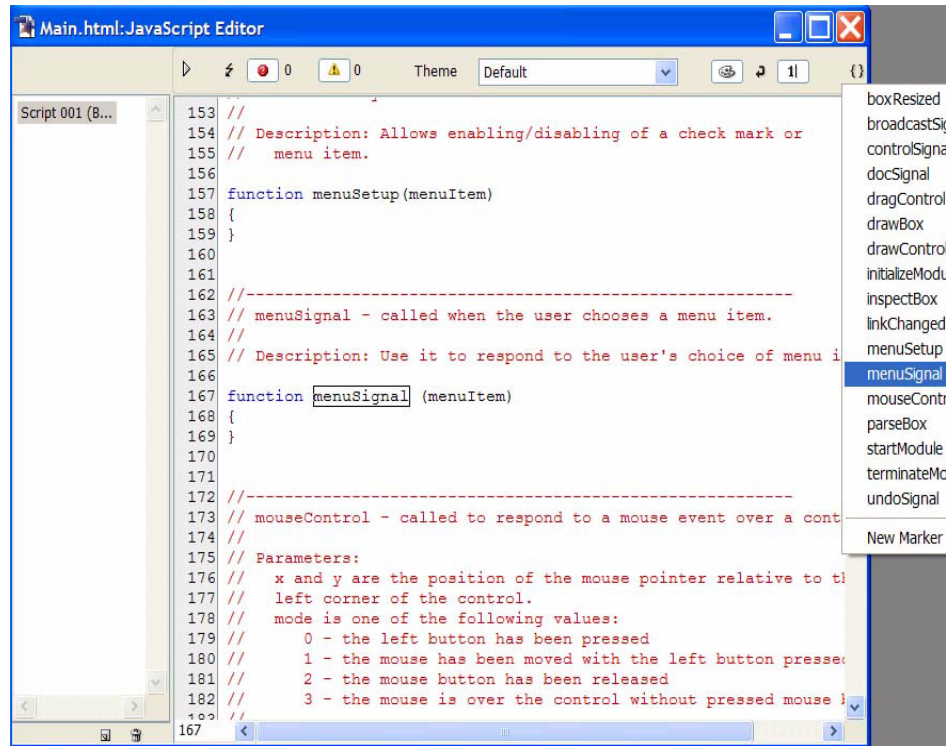
3. Scroll down the the bold word **Scripts**. Double click on the JavaScript icon below Scripts.



This opens up the JavaScript code for the extension.

4. Click on the Navigate through code { } to list the functions contained in the JavaScript file.

5. Click on `menuSignal()` to jump to this function in the JavaScript editor window.

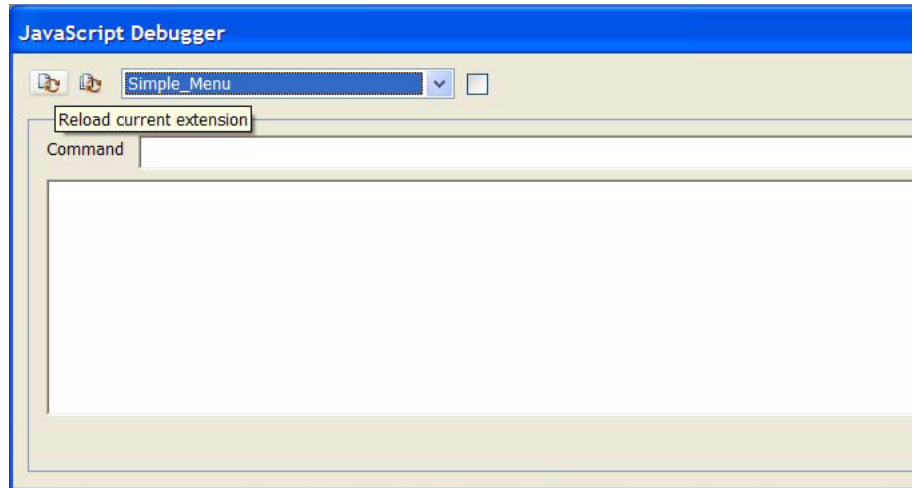


6. Add the following JavaScript code to `menuSignal()`:

```
function menuSignal (menuItem)  
{  
    // new code begins here  
    writeln("menuItem.name: " + menuItem.name);  
    if (menuItem.name == "SimpleMenuItem") {  
        alert("Simple Menu Item selected.");  
    }  
    else {  
        alert("Something went wrong.");  
    }  
    // new code ends here  
}
```

7. Close the JavaScript editor.
8. Save Main.html.

9. Open the JavaScript Debugger from **Special > JavaScript Debugger**. From the pull down menu, select **Simple Menu** (extension name). Reload changes dynamically by selecting **Reload current extension** in the JavaScript Debugger.



10. Select **Extensions > Simple Menu Item**.
If everything went correctly, the alert “Simple Menu Item selected” appears.

4.0 Extra Credit: using the Dialog Editor

Had enough? Then skip on to [Section 6.0: Next steps](#). This extra credit section guides you through the steps to design and add a dialog to the Simple Menu extension.

You installed the Dialog Editor extension in [Section 1.1: GoLive installation](#). The Dialog Editor contains tools to easily create and layout dialog boxes for use in your extensions.

4.1 Design the dialog

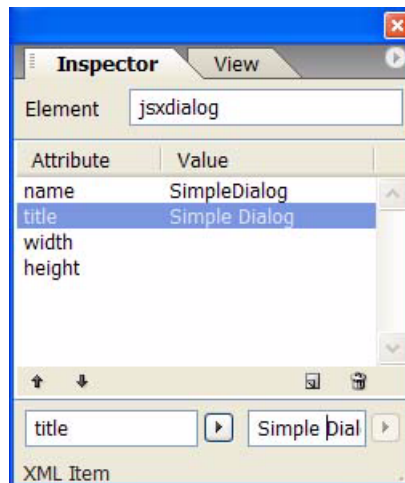
Follow these steps to layout the dialog.

1. If the **Main.html** file for Simple Menu is not open, select **Extensions > Edit Extensions...** Select **Simple Menu**, and click **OK**.
2. In **Main.html** for Simple Menu, scroll down until you see the bold word **Dialogs**.
3. Select the comment below **Dialogs**, and with the right mouse button, select **Unwrap Comment** **Content** from the pulldown menu. If you are on Macintosh, select the

comment, and control click to get the pull-down menu. When the comment is unwrapped, the `<jsxdialog>` tag appears.

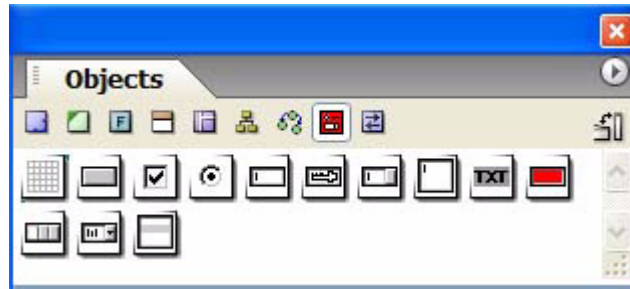


4. Select `jsxdialog`, and in the Inspector set Name to `SimpleDialog`; set Title to Simple Dialog.



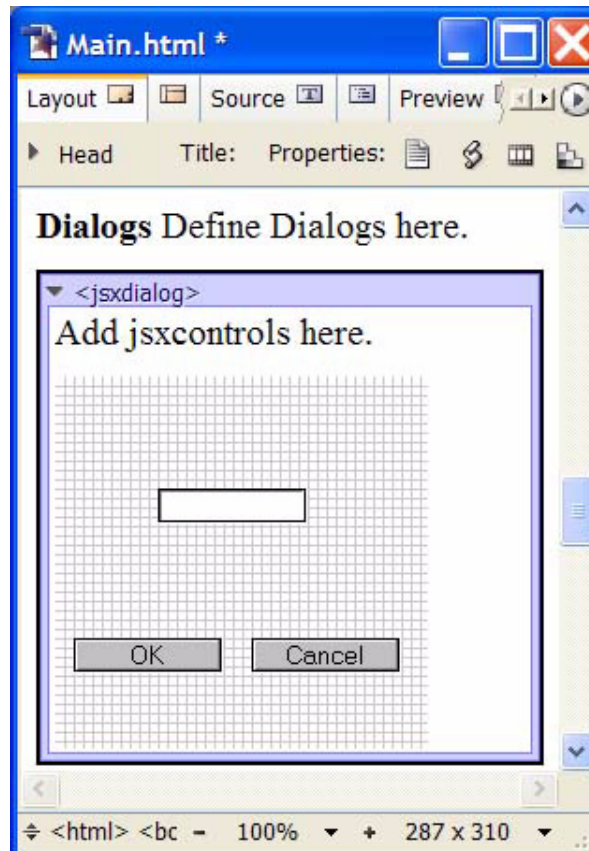
5. In `jsxdialog`, click on the right pointing arrow to expand the box. You will see the words: Add `jsxcontrols` here.

6. In the Objects Palette, select the Dialog Editor tab.



7. In the Dialog Editor tab, drag the Dialog Grid object inside the jsxdialog object.
8. In the Inspector, note the height and width of the Layout Grid. In the Inspector set the height and width attributes of the jsxdialog object to the same values.
9. Drag an Edit Field from the Objects Palette onto the page. In the Inspector for Edit Field, set the Width to 120, and remove the "Edit" in Value.
10. Drag a Button from the Objects Palette onto the page. In the Inspector for Button, change Value to OK.

11. Drag another Button onto the page just to the right of the OK button. In the Inspector, change Name to dialogcancel (no capitalization), and change Value to Cancel.



4.2 Add JavaScript code to display dialog

Now that the dialog has been designed, we'll add the code to display the dialog.

1. Scroll up to the bold word **Scripts**. Double click on the JavaScript icon below Scripts to open the JavaScript editor.
2. Click on `menuSignal()` from the Navigate through code `{ }` list, to jump to this function in the JavaScript editor window. We will modify this function as follows:

```
function menuSignal (menuItem)
{
    writeln("menuItem.name: " + menuItem.name);
    if (menuItem.name == "SimpleMenuItem") {
        // here's the new code:
        if (SimpleDialog.runModal()) {
            // display the value of the SimpleDialog's
            // edit box
            alert(SimpleDialog.edit.value);
        }
        else {
```

```
        alert("User selected Cancel.");
    }
    // new code end
}
else {
    alert("Something went wrong.");
}
}
```

3. Since we added a static `jsxdialog` element it cannot be reloaded dynamically like scripts or other dynamically created objects. Hence close the JavaScript editor, save `Main.html`, and restart GoLive.

4.3 Display the dialog

In the last step, you restarted GoLive. Now let's test out the dialog.

1. From the **Extensions** menu, select **Simple Menu Item**.
2. Your Simple Dialog should appear.



3. Type something in the edit box, and click **OK**.
Whatever you typed appears in an alert.

5.0 Troubleshooting

While installing or creating extensions for GoLive, problems may arise due to programming mistakes or HTML errors. If GoLive begins acting strangely, try one or both of the following:

- Disable the extension in the **Edit > Preferences, Modules > Extend Scripts** dialog.
- Remove the extension folder from the `GoLive/Modules/Extend Scripts` folder.

6.0 Next steps

Congratulations! You've just completed your first GoLive extension.

Next steps

To learn more about the GoLive Extend Script SDK, the Adobe GoLive SDK 8.0r1 folder contains documentation, tools, and samples.

The Adobe Studio Exchange (<http://www.adobetudioexchange.com>) is a site where you can download many useful extensions created by our developer community.

© 2005 Adobe Systems Incorporated. All rights reserved.

Adobe, Adobe logo and GoLive are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Macintosh is a trademark of Apple Computer, Inc., registered in the United States and other countries. Java, JavaScript and all Java-related marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.