

GoLive CS2 SDK Release Notes

This document describes new features and changes in the GoLive CS2 SDK (version 8.0r1) and summarizes the available samples.

- [New in GoLive CS2 SDK Release 8.0r1](#)
- [SDK Samples](#)

New in GoLive CS2 SDK Release 8.0r1

New Features

The GoLive CS2 SDK release adds significant new features to the GoLive SDK. This version provides:

- Access to CSS style sheets through the GoLive DOM. New objects allow you to create, delete and maintain both internal and external style sheets associated with documents, and edit the style sheets themselves as a new document type in GoLive.
- A new event handling mechanism, Dynamic Event Handling, which replaces the previous callback mechanism. Your script can register one or more handler functions for a user-interaction event. Events in the Site window that were previously handled separately are now handled by the same mechanism as all other events.

The older event mechanism that uses callbacks is still supported, but it is recommended that you switch to the new one.

- Access to embedded XMP metadata for a file, through a new XMP object, with functions to access, set, and delete properties in the metadata.
- The ability to create your own interface to any version control system, using the new `VCSHandler` object and your own handlers for new version-control events.
- Limited access to the content placed in the platform's clipboard through the new `app.clipboard` property.
- Enhancements to the user-interface controls, including:
 - The ability to dock and collapse palettes.
 - Better control and access to the sizes and positions of windows and controls.
 - The new panel type `splitpanel`, which contains two subpanels that can be arranged vertically or horizontally. The border between the panels can be moved with the mouse.
 - The new control type `hierarchy`, which replaces the old `tree` control type.
 - The new file-selection control type `filelist`, a tree that displays files and folders for a selected local or remote filesystem location, and allows the user to make multiple selections.
 - The ability to specify tooltips for controls.
- The ability to extend context menus by adding submenus or menu items programmatically. A new set of events allows you to handle interaction with context menus.
- Additional site management features, including:
 - The ability to create a new site.

- A new site event that allows you to respond to a user beginning an upload of site files.
- The ability to modify `siteReference` collection.
- Properties for accessing a site's folder structure.
- A new method, `document.createPDF()`, that allows you to write out a document's content in PDF format.

Updated Documentation

The GoLive CS2 SDK documentation has been updated to include the new features and mechanisms used by version 8.0, and information on previously available features has been updated.

- The *SDK Programmer's Guide* contains instructions and recommendations for how to use the SDK to create extensions.
- The *SDK Programmer's Reference* contains complete descriptions and syntax details of the SDK objects, properties, and functions, of the SDK-defined JavaScript tags, and of various related syntax schemes.

Both books are intended for developers who are knowledgeable programmers, who understand HTML and JavaScript concepts, and who are familiar with Adobe GoLive CS2. The books are companion volumes, with many cross-references to aid you in finding both the general and specific information you need for any development task. To make the cross-references work as live links between the documents, keep both PDF files open in Adobe Acrobat or Adobe Reader.

SDK Samples

The SDK offers a large library of samples that demonstrate features and development techniques.

New Samples

The following samples have been added to the 8.0r1 release.

- BridgeSample
- CrossDOM
- XMP

Changes in Samples

- The Custom Box sample has been updated to use the new event-handling mechanism.
- The FolderTreeObject sample has been updated to use the new `hierarchy` control type, rather than the deprecated `tree` control type.

Samples Summary

This section describes the samples included in this version of the GoLive CS SDK. The SDK samples are organized into three folders:

- [C Library Samples Folder](#)
- [Samples Folder](#)
- [Tools Folder](#)

The core samples (marked with *) are referenced in the documentation and may be used as a starting point for learning the SDK and developing your own extensions

C Library Samples Folder

This folder contains samples and include files for building C Libraries. See the *GoLive CS SDK Programmer's Reference* for more information.

Binary API*	Shows how to write an external library in C. External libraries can be made available to all SDK extensions. This sample draws a diagram and counts powers of numbers using C library functions.
changeCaps	Adds a menu item, Change Caps, that accesses various functions to change the case of the text selection.
changeCaps binary	Uses C functions to change the case of text sent to it by the changeCaps extension. These functions are referenced by the changeCaps sample.
Custom Windows Dialog (Windows only)	Shows how to use the MFC classes to integrate a Windows modal dialog into a GoLive Extension. This is referenced by the Windows Dialog extension to display a calendar in GoLive.

Samples Folder

The following samples are contained in the `Samples` folder.

Apply Style	Applies style attributes to all <code>textArea</code> objects in the documents.
BridgeSample	Shows how to use the messaging framework to communicate and exchange data between Adobe Bridge and GoLive. (For details of the messaging framework, see the <i>Bridge Scripting Guide</i> , available with Creative Suite 2.)
chgBkg_List_OpenPage	Changes the background color of all the pages in your Web site. It first builds a list of the pages and then processes the list; it opens each page, changes the markup tree, and reparses the page so that the change becomes visible. This code is easily modified to make other changes to all the pages in your Web site. This extension is provided only as an example of using the SDK. It is not the recommended way to change the background color of all the pages in your Web site, since GoLive already offers that capability using the Elements tab of the Find command's window.
chgBkg_OpenMarkup	Changes the background color of all the pages in your Web site using <code>openMarkup()</code> and processing pages as they are found. It makes a list of all the pages in the site and then processes them all at once, opening each page. This version is much faster than <code>chgBkg_List_OpenPage</code>. This code is easily modified to make other changes to all the pages in your Web site.
CrossDOM	Shows how to use the cross-DOM functions defined by all Creative Suite 2 applications to communicate with other applications. (For details of the cross-DOM feature, see the <i>Bridge Scripting Guide</i> , available with Creative Suite 2.)
Custom Box*	Implements a custom tag that has an entry in the Objects palette and adds a custom HTML element to the page when the user drops it on a GoLive document window. Also provides a custom Inspector window that enables the user to modify the properties of the custom element interactively. This also demonstrates the dynamic localization features.
CSV MLGenerator	Provides additional markup generators for components and templates.
DialogExtract	Extracts controls from a dialog created using the Dialog Editor* .
Document Events*	Displays the paths of all the open documents and helps distinguish between them (as all extension source files have the same name, <code>Main.html</code>). It uses the document events to keep track of documents.
Document Source*	Shows how to access the document source view.
DragnDrop*	Shows how to use the <code>dragDropEvent</code> object to handle drag-and-drop actions. Also shows drag-and-drop with custom controls when you want to drop html or colors or files on a custom control.
Dynamic UI*	Shows how to dynamically create most of the GoLive UI elements, such as palettes, dialogs, inspectors, buttons, preview object, a new document tab, tab views in a palette, and more.

Encoding	Changes the encoding for all comments in a document.
FolderTreeObjects	Shows how to use the <code>tree</code> control object. Adds a Folder Tree palette to the Window menu. This palette displays the heirarchy of all folders and files in the folder selected by user.
FormLikeTag	Shows how to make a GoLive container tag similar to the <code><form></code> tag. This extension can be used as a starting point if you need to create container tags. GoLive container tags are most often used to support middleware or database tags. Also shows the rollover icons for the mouse-over state for a palette entry using its <code>hilitepicture</code> attribute.
FTP and WebDAV*	Shows the use of the <code>FTP</code> and <code>WebDAV</code> objects. Allows you to copy files to or from these servers.
ImageConverter*	Displays a palette which converts all files in a folder to the specified Web-safe image formats such as GIF and JPEG. Can convert Illustrator®, PhotoShop® files and other Smart Objects. Also shows the use of <code><jsxview></code> tag and view resources.
KeyMap*	Displays a palette that allows the user to insert a character in a document by clicking on the character map. Uses external images to customize its display according to the current hardware platform.
Layout	Demonstrates the use of the <code>layout</code> object to modify a table.
Markup Tree*	Demonstrates how to work with the document markup tree.
Menus and Dialogs*	Creates custom menus, menu items, dialogs, and dialog controls. Demonstrates the dynamic localization features. Also calls the TimerPalette to show inter-extension communication.
Multiple Inspectors	Shows a potential use of multiple inspectors for an object. This sample switches the inspector based on whether the document is HTML, WML, or SMIL.
Multiple URL	Shows how to implement a custom box with an inspector that contains multiple <code>urlgetter</code> controls with undo support.
Palettes*	Shows how to implement a floating palette window and palette menu. Also calls a function defined by the Binary API sample.
Password	Demonstrates the use of the <code>password</code> control. Defines a login dialog that allows the user to set a User Name and Password.
Preferences	Shows how to programatically set and get preferences for the GoLive application. These are the preferences accessible from the Edit > Preferences menu. This sample has been changed to use the latest techniques; it no longer modifies the <code>prefs</code> object.

RedefineTag	Shows a how to redefine a standard GoLive tag for all instances of that tag, display a custom box associated with the standard tag, add custom attributes, and create a customized Inspector. Do not enable both the RedefineTag and RedefineTagSelective extensions in the same GoLive session. If both extensions are loaded and you switch to source view and back, the image tag or redefined image tag becomes invisible.
RedefineTagSelective	Shows a how to redefine one instance of a standard GoLive tag, display a custom box associated with that tag, add custom attributes, and create a customized Inspector. (See note for RedefineTag.)
RedefineTagWithGlue	Demonstrates the use of the <code>glue</code> attribute to selectively define an object and inspector based on document type.
Register Callback	Demonstrates how to define a custom callback.
Settings	Demonstrates how to work with a <code>settings</code> object.
SetTitle	Allows the user to set the title of the document when opening a new file. Also checks the title on open and save to see if it is set to the default "Welcome to Adobe GoLive", and prompts the user to set a title.
Site Navigation	Shows how to programatically traverse the files in a site.
Sort	Sorts a list.
Table Properties	Changes table properties on all tables in a document.
textArea	Shows how to get all <code>textArea</code> objects in a document.
TimerPalette	Defines a palette that can be used as a stopwatch or timer. Called from Menus and Dialogs* to show inter-extension communication.
Translate SSI	Demonstrates the <code><jsxtranslator></code> tag and translator functionality. Includes an HTML snippet in a file using the SSI include directive.
Translate Time	Demonstrates the <code><jsxtranslator></code> tag and translator functionality. Defines a tag <code><time></code> that inserts the current time in the document.
URLize	Adds a menu with two items that convert a list of text items in email, URL, or FTP format into links. For example: <pre>http://www.adobe.com someone@adobe.com</pre> are converted to: <pre><ahref="http://www.adobe.com">http://www.adobe.com <ahref="someone@adobe.com">someone@adobe.com</pre>
WebSite	Demonstrates the <code>webSite</code> object.
WebSite Columns	Uses the <code>webSite</code> object to create custom columns in the Site window. Modified to show the use of <code>siteTabsEvent</code> object.
WebSite Export	Uses the <code>webSite</code> object to export and cleanup a site. This object offers export settings similar to the menu items File > Export > Site and Site > Clean Up Site .

Windows Dialog (Windows only)	Shows inter-application communication between a Windows MFC dialog and the Adobe GoLive application. Displays the Windows MFC modal dialog with a calendar control in GoLive. Uses the WinDlg C Library sample to display the dialog.
XMP	Creates a palette in which the user can display and modify a file's XMP metadata using the XMP object.

Samples/Common Folder

The files in the `Common` folder are binary libraries (.dll on Windows) and JavaScript files (.js). When installed in the `GoLive/Modules/Extend Scripts/Common` folder, these files provide common object and functions that are available to all extensions.

changeCaps(.dll)	The binary library from the C Library changeCaps binary sample. Its functions are used by the changeCaps sample.
JSASample(.dll)	The binary library from the C Library Binary API* sample. Its functions are used by the Menus and Dialogs* sample extension.
WinDlg(.dll) (Windows only)	The binary library from the Custom Windows Dialog (Windows only) sample. Its functions are used by that sample extension.

Samples/Shared/Adobe Folder

The `Shared` folder contains general purpose JavaScript functions shared between multiple extensions. Include these files in your extension by adding a script tag with a `src` attribute referencing a file in the shared folder:

```
<script src="../../Shared/Adobe/generalUtils.js">
```

generalUtils.js	General utility functions.
markupUtils.js	Utilities for working with the markup tree.

Tools Folder

These tools are provided to assist in developing extensions:

Dialog Editor*	Adds a tab to Objects palette with several custom elements for controls that are useful in dialogs. You can drag these palette entries to a GoLive document window to design a custom dialog for your extension.
Edit Extension	Allows you to easily edit the <code>Main.html</code> file (source code) for an extension that is installed in GoLive.
Extension Builder	Helps to create a new extension by creating a new <code>Main.html</code> file and saving it in a directory of your choice. Uses the file <code>Skeleton.html</code> as a template. You can customize this template for your own needs.