

Scripting Read Me for Adobe® InDesign® CS4

This document contains information about scripting in Adobe InDesign CS4, including:

- A summary of the InDesign CS4 scripting documentation (see [“InDesign Scripting Documentation” on page 1](#)).
- Directions for running a script (see [“Running Scripts” on page 2](#)).
- A list and brief description of InDesign sample scripts (see [“Sample Scripts” on page 2](#)).
- A list of known issues in InDesign CS4 scripting (see [“Known Issues Related to InDesign CS4 Scripting” on page 9](#)).

For more information on InDesign scripting, go to the InDesign scripting home page; <http://www.adobe.com/products/indesign/scripting/index.html>, or visit the InDesign scripting User-to-User forum., <http://www.adobeforums.com>.

For late-breaking InDesign scripting news, see the latest version of this file on the InDesign Scripting home page.

InDesign Scripting Documentation

InDesign CS4 scripting documentation can be downloaded from the InDesign scripting home page (listed above). The scripting documentation set for InDesign CS4 comprises the following:

- *Adobe InDesign CS4 Scripting Tutorial* — Shows how to get started with InDesign scripting. Covers AppleScript, JavaScript, and VBScript in one PDF document. The introductory scripts in this document are available as a single ZIP archive or can be copied from the PDF.
- *Adobe InDesign CS4 Scripting Guide* (AppleScript, JavaScript, and VBScript versions) — Discusses more advanced InDesign scripting topics. All tutorial scripts shown are included in a single ZIP archive, so there is no need to copy and paste scripts from the PDF. (Most scripts shown in the text are incomplete fragments demonstrating a specific property, method, or technique.)
- *JavaScript Tools and Features* — Covers using the ExtendScript Toolkit for JavaScript development, creating user interfaces with ScriptUI, using the File and Folder objects, and other features specific to the ExtendScript language (Adobe’s version of JavaScript).

There is no *Scripting Reference* PDF for CS4; instead, use the object-model viewer included with your script-editing application (as described in *Adobe InDesign CS4 Scripting Tutorial*).

InDesign sample scripts are installed by default. They appear in the Scripts panel (Window > Automation > Scripts).

Installing the scripting documentation scripts

In addition to the sample scripts, all scripts shown (in fragmentary form) in the scripting documentation are available for download from the InDesign scripting home page.

After downloading the script ZIP archive, follow the steps below to extract the scripts from the archive. Once you extract the files, move the folder(s) for the language(s) you want to work with (AppleScript, JavaScript, and/or VBScript) to your Scripts Panel folder. (For more on installing scripts, see the *Adobe InDesign CS4 Scripting Tutorial*.)

Windows

With a standard Windows installation, you can extract the ZIP files using the Windows Extraction Wizard, as described below. If you installed a third-party ZIP application (like WinZip), you may have a customized menu instead of the menu items listed below.

1. Copy the ZIP file to your hard drive.
2. Right-click the file, then choose Extract All...
3. When presented with the Extraction Wizard dialog, click Next.
4. Choose a location for your files or use the default location (the current directory).
5. When the extraction is complete, click Finish.

Mac OS

1. Copy the ZIP file to your hard drive.
2. Double-click the ZIP file.

Running Scripts

To run a script, double-click the script name in the Scripts panel.

On Windows, you can run VBScripts (file extension `.vbs`) or JavaScripts (file extension `.jsx`). On Mac OS, you can run AppleScripts (file extension `.applescript`) or JavaScripts.

Sample Scripts

InDesign CS4 includes the sample scripts listed below.

Script name	For more information, see page ...
AddGuide	3
AddPoints	4
AdjustLayout	4
AlignToPage	4
BreakFrame	4
CornerEffects	5
CreateCharacterStyle	5

Script name	For more information, see page ...
CropMarks	5
ExportAllStories	6
FindChangeByList	6
ImageCatalog	6
LabelGraphics	6
LabelGraphicsMenu	7
MakeGrid	7
Neon	7
PathEffects	8
PlaceMultipagePDF	8
SelectObjects	8
SortParagraphs	8
SplitStory	9
TabUtilities	9

Make sure you save your work before running a sample script for the first time.

Before using sample scripts on important InDesign documents, experiment with them so you understand what they do.

Script Descriptions

This section includes a brief description of each sample script.

AddGuides

Adds guides around the selected object or objects.

Demonstrates:

- Processing objects in the selection.
- Getting page coordinates from objects.
- Adding guides to a spread.
- Creating a layer (and checking whether a layer already exists).
- Creating a user interface. This example includes both a dialog box created using the InDesign dialog object and an optional dialog box created using the ExtendScript ScriptUI features.

AddPoints

Adds points to the paths of the selected object or objects.

Demonstrates:

- Processing objects in the selection.
- Bezier math and path-point manipulations.
- Using the “entire path” property.
- Creating a user interface.

AdjustLayout

Moves objects by specified distances on right/left pages.

Demonstrates:

- Using the move method to move objects.
- Working with page ranges.
- Determining whether a page is a right or left page.
- Creating a user interface.

AlignToPage

Aligns objects to specified positions on a page.

Demonstrates:

- Using the move method to move objects.
- Working with page sizes and page margins.
- Creating a user interface.

BreakFrame

Removes a selected text frame and its contents from a story.

Demonstrates:

- The duplicate method.
- Working with text-frame contents.

NOTE: BreakFrame does not work properly when the text frame contains a table that is continued from or continues to another text frame.

CornerEffects

Re-draws the path of the selected item(s) using a variety of corner effects. Corner effects can be applied to selected points on the path.

Demonstrates:

- Working with the selection.
- Mathematical operations for simple path operations.
- Using the “entire path” property.
- Creating a user interface.

CreateCharacterStyle

Defines a complete character style based on the selected text.

Demonstrates:

- Processing objects in the selection.
- Creating a character style.
- Getting text-formatting attributes from a text object.
- Filling in character-style properties.

CropMarks

Adds crop and/or registration marks around the selected object or objects.

Demonstrates:

- Processing objects in the selection.
- Getting page coordinates from objects.
- Finding the bounding box of a series of selected items.
- Drawing graphic lines and ovals.
- Applying swatches (and the “Registration” color).
- Creating a layer (and checking whether a layer already exists).
- Creating a user interface.

ExportAllStories

Exports all stories in a document to a series of text files.

Demonstrates:

- Exporting text.
- JavaScript file/folder objects and methods (JavaScript only).
- Creating file names based on ID attributes.
- Creating a user interface.

FindChangeByList

Performs a series of common text find/change operations by reading a tab-delimited text file.

For more information, see the start of the script or the start of the FindChangeList.txt file. You can add your own find/change operations to the FindChangeList.txt file, or create your own file.

Demonstrates:

- Working with files and folders.
- Finding a file/folder relative to the active script.
- Using the text find/change methods (text, grep, and glyph).
- Reading tab-delimited text data from a text file.

ImageCatalog

Places all graphics in a specified folder in a “contact sheet” layout.

Demonstrates:

- Working with files and folders.
- Placing and fitting graphics.
- Creating a text frame and entering text.
- Creating a user interface.

LabelGraphics

Adds a caption (containing the filename, file path, XMP description, or XMP author) next to every imported graphic.

Demonstrates:

- Working with links.
- Getting XMP data from imported graphics.

- Creating a text frame and entering text.
- Creating a user interface.

LabelGraphicsMenu

Creates a menu item on the Layout context menu. You can use this menu item to add a caption (containing the filename, file path, XMP description, or XMP author) next to the selected graphic(s).

Demonstrates:

- Adding a script menu item to a menu.
- Using a “beforeDisplay” handler to show/hide the menu item depending on the selection.
- Working with links.
- Getting XMP data from imported graphics.
- Creating a text frame and entering text.
- Creating a user interface.

Associated scripts

The VBScript and AppleScript versions of the LabelGraphicsMenu script look in the folder containing the script for a folder named LabelGraphicSupport. Inside this folder, the script requires:

- LabelGraphicBeforeDisplay
- LabelGraphicOnInvoke

The folder also includes the RemoveLabelGraphicMenu script, which can be used to delete the menu item.

MakeGrid

Creates a grid by subdividing or duplicating the selected object or objects.

Demonstrates:

- Duplicating and moving objects.
- Creating a user interface.

Neon

Applies a “blend” effect to the selected object or objects.

Demonstrates:

- Duplicating an object.
- Applying a stroke color.
- Grouping objects.
- Applying graphic-formatting attributes (color, stroke weight, tint).
- Creating a user interface.

PathEffects

Changes the position of path points on the selected object or objects to add a creative effect.

Demonstrates:

- Processing objects in the selection.
- Duplicating an object.
- Bezier math and path-point manipulations.
- Simple trigonometry.
- Creating a user interface.

PlaceMultipagePDF

Places all pages of a PDF.

Demonstrates:

- Working with PDF attributes.
- Placing graphics.
- Adding pages.
- Selecting a file.
- Creating a user interface.

SelectObjects

Selects objects on the active spread by their object type.

Demonstrates:

- Basic page-item traversal.
- Selection.
- Creating a user interface.

SortParagraphs

Sorts the paragraphs in the selection alphabetically.

Demonstrates:

- Text-object move method.
- Simple bubble sort.

SplitStory

Splits the text frames in the selected story into separate, unlinked text frames.

Demonstrates:

- Duplicating objects.
- Deleting objects.
- Iterating through each text frame in a story.

NOTE: SplitStory does not work properly when a text frame in the story contains a table that is continued from or continues to another text frame.

TabUtilities

Applies tab stops and indents to the selected text.

Demonstrates:

- Working with tab stops and indents.
- Getting page positions from text objects.
- Getting the text column containing the cursor.
- Creating a user interface.

Known Issues Related to InDesign CS4 Scripting

JavaScript start-up scripts

User start-up scripts should be placed in the InDesign start-up scripts location (where they will run once each time the application is launched), not in the ExtendScript engine initialization scripts location (where they will run each time an engine is initialized).

To run scripts when InDesign starts, put them in the `Startup Scripts` folder inside the `Scripts` folder in your `InDesign` folder. (Create this folder if it does not already exist.)

Cannot set the midpoint location for an opacity gradient stop

If you try to set the midpoint location for the first opacity gradient stop in a gradient feather settings object, InDesign returns an error.

Scripts run outside InDesign cannot create persistent ExtendScript engines (JavaScript only)

As discussed in Chapter 2, “Scripting,” of *Adobe InDesign CS4 Scripting Guide: JavaScript*, ExtendScript scripts can create persistent instances of the ExtendScript engine. Functions and variables defined in the persistent engine can be used by other scripts that execute in that engine. To create a persistent ExtendScript engine, however, the script must be run from the InDesign Scripts panel—running the script

from the ExtendScript Toolkit or via BridgeTalk from another application does not create the persistent engine.

Event listeners added or removed during event propagation are not handled according to the W3C specification

The *W3C Document Object Model (DOM) Level 2 Events Specification* (see <http://www.w3.org/TR/DOM-Level-2-Events/Overview.html>) states:

“If an `EventListener` is added to an `EventTarget` while it is processing an event, it will not be triggered by the current actions but may be triggered during a later stage of event flow, such as the bubbling phase.

...

If an `EventListener` is removed from an `EventTarget` while it is processing an event, it will not be triggered by the current actions. `EventListeners` can never be invoked after being removed.”

In InDesign scripting, event listeners added to an event target during event propagation are not triggered for the duration of the event. Event listeners removed from an event target during event propagation are triggered by the event (i.e., the event listeners are removed when event processing is complete,).

The ExtendScript toolkit does not show a list of InDesign scripts (Mac OS only)

By default, the ExtendScript Toolkit does not specify a target application when it starts. This means the list of available scripts in the Scripts panel (in the ExtendScript Toolkit, not in InDesign) is not populated with available InDesign scripts. Set the target application to InDesign, and the ExtendScript Toolkit populates the Scripts panel.

Documents must be saved before packaging

You must save a document using before using the `package` method. If you do not save the document, InDesign generates an error.