Scripting Read Me for Adobe® InDesign® CS3

This document contains late-breaking information about scripting in Adobe InDesign CS3, including:

- Important changes in scripting from InDesign CS2 to InDesign CS3 (see "Changes in InDesign Scripting" on page 1).
- Corrections to *Adobe InDesign CS3 Scripting Guide* (see "Scripting Documentation Corrections" on page 6).
- Known issues (see "Known Issues Related to InDesign CS3 Scripting" on page 6).

Changes in InDesign Scripting

Some aspects of InDesign scripting have changed since InDesign CS2, including the following:

- The scripting documentation was reorganized (see "Scripting Documentation Reorganization" on page 1).
- The `place` method now returns an array, rather than a single object (see "The Place Method Now Returns an Array, Rather Than a Single Object" on page 2).
- The `resize`, `rotate`, and `shear` methods were replaced by a single method, `transform`. (see "Transforming Objects" on page 2).
- The order of the items in the `selection` property has changed.

Note: Remember you can run your InDesign CS2 (version 4.x) and CS (version 3.x) scripts using the script-versioning feature. For more information on script versioning, see Chapter 2, "Scripting Features," in *Adobe InDesign CS3 Scripting Guide*.

Scripting Documentation Reorganization

The scripting documentation set for InDesign CS3 differs from the documentation set for InDesign CS2. In InDesign CS3, the scripting documentation comprises the following:

- *Adobe InDesign CS3 Scripting Tutorial* — Shows how to get started with InDesign scripting. Covers AppleScript, JavaScript, and VBScript in one PDF document. The introductory scripts presented in this PDF are available as a single Zip archive or can be copied from the PDF.
- *Adobe InDesign CS3 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 this release. Instead, use the object-model viewer included with your script-editing application (as described in *Adobe InDesign CS3 Scripting Tutorial*).

InDesign sample scripts are now installed by default.

Installing the Scripting Documentation Scripts

Perform the following steps to extract the scripts from the Zip archive. Once you extract the files, move the folder for the language you want to work with (AppleScript, JavaScript, or VBScript) to your Scripts Panel folder. (For more on installing scripts, see *Adobe InDesign CS3 Scripting Tutorial*.)

Mac OS

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

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), however, 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 the Extract All... menu item.
3. When presented with the Extraction Wizard dialog, click the Next button.
4. Choose a location for your files, or use the default location (the current directory).
5. When the extraction is complete, click the Finish button.

The Place Method Now Returns an Array, Rather Than a Single Object

In InDesign CS2 scripting, the `place` method returned a single object. In InDesign CS3 scripting, the same method returns an array, because some forms of the `place` method can place multiple files (just as you can using the InDesign user interface).

Transforming Objects

In InDesign CS2 scripting, you used the `resize`, `rotate`, and `shear` methods to transform page items. In InDesign CS3, those methods were replaced with a single method, `transform`. The `transform` method uses a `transformation matrix` object to apply all transformations (move, scale, shear, and rotate) to an object. Multiple transformations may be applied in a single operation.

In addition, the repeated transformations available in the InDesign user interface are now supported by scripting: `transform again`, `transform again individually`, `transform sequence again`, and `transform sequence again individually`. You also can remove all transformations applied to an object, using the `clear transformations` method.

The following examples show how to perform common transformations using the `transform` method.

AppleScript

```
--Transform.applescript
--An InDesign CS3 AppleScript
--
--A brief demonstration of the new transform method.
--
--The transform method has the following parameters:
--
--Coordinate space (pasteboard coordinates, parent coordinates, or inner
coordinates)
--The coordinate space to use for the transformation.
--
--From (as an array of two numbers, an anchor point enumeration,
--or a bounding box limits enumeration):
--The center of transformation.
--
--With Matrix (Transformation Matrix):
--Specify the matrix used to transform the object.
--
--Replacing Current (Boolean):
--If true, replace the current transformation values of the object
--with the values of the transformation matrix.
--
--Considering Ruler Units (Boolean)
--If true, use the current ruler usings and zero point location
--when transforming the object. If false, use the coordinates
--relative to the Coordinate Space parameters.
--
tell application “Adobe InDesign CS3”
   set myDocument to make document
   set myRotateMatrix to make transformation matrix with properties
{counterclockwise rotation angle:27}
   set myScaleMatrix to make transformation matrix with properties {horizontal
scale factor:0.5, vertical scale factor:0.5}
   set myShearMatrix to make transformation matrix with properties {clockwise
shear angle:30}
   tell page 1 of myDocument
      --Create an example page item.
      set myRectangle to make rectangle with properties {geometric bounds:{“72pt”,
“72pt”, “144pt”, “288pt”}}
      --Rotate the page item.
      transform myRectangle in pasteboard coordinates from center anchor with
matrix myRotateMatrix
      --Create another example page item.
      set myRectangle to make rectangle with properties {geometric bounds:{“72pt”,
“72pt”, “144pt”, “288pt”}}
      --Scale the page item.
      transform myRectangle in pasteboard coordinates from center anchor with
matrix myScaleMatrix
      --Create another example page item.
      set myRectangle to make rectangle with properties {geometric bounds:{“72pt”,
“72pt”, “144pt”, “288pt”}}
      --Shear the page item.
      transform myRectangle in pasteboard coordinates from center anchor with
matrix myShearMatrix
   end tell
end tell
```

JavaScript

```
//Transform.jsx
//An InDesign CS3 JavaScript
//
//A brief demonstration of the new transform method.
//
//The transform method has the following parameters:
//
//Coordinate Space (CoordinateSpaces.pasteboardCoordinates,
//CoordinateSpaces.parentCoordinates, or CoordinateSpaces.innerCoordinates)
//The coordinate space to use for the transformation.
//
//From (as an array of two numbers, an AnchorPoint enumeration,
//or a BoundingBoxLimitsEnumeration):
//The center of transformation.
//
//With Matrix (Transformation Matrix):
//Specify the matrix used to transform the object.
//
//Replacing Current (Boolean):
//If true, replace the current transformation values of the object
//with the values of the transformation matrix.
//
//Considering Ruler Units (Boolean)
//If true, use the current ruler usings and zero point location
//when transforming the object. If false, use the coordinates
//relative to the Coordinate Space parameters.
//
var myRotateMatrix =
app.transformationMatrices.add({counterclockwiseRotationAngle:27});
var myScaleMatrix = app.transformationMatrices.add({horizontalScaleFactor:.5,
verticalScaleFactor:.5});
var myShearMatrix =app.transformationMatrices.add({clockwiseShearAngle:30});
var myDocument = app.documents.add();
var myRectangle =
myDocument.pages.item(0).rectangles.add({geometricBounds:["72pt", "72pt",
"144pt", "288pt"]});
//Template for rectangle.transform():
//transform(in as CoordinateSpaces, from, withMatrix, [replacingCurrent],
[consideringRulerUnits as Boolean = False])
myRectangle.transform(CoordinateSpaces.pasteboardCoordinates,
AnchorPoint.centerAnchor, myRotateMatrix);
//Create another rectangle.
myRectangle = myDocument.pages.item(0).rectangles.add({geometricBounds:["72pt",
"72pt", "144pt", "288pt"]});
//Transform the rectangle.
myRectangle.transform(CoordinateSpaces.pasteboardCoordinates,
AnchorPoint.centerAnchor, myScaleMatrix);
//Create another rectangle.
myRectangle = myDocument.pages.item(0).rectangles.add({geometricBounds:["72pt",
"72pt", "144pt", "288pt"]});
//Transform the rectangle.
myRectangle.transform(CoordinateSpaces.pasteboardCoordinates,
AnchorPoint.centerAnchor, myShearMatrix);
```

VBScript

```
Rem Transform.vbs
Rem An InDesign CS3 VBScript
Rem
Rem A brief demonstration of the new transform method.
Rem
Rem The transform method has the following parameters:
Rem
Rem Coordinate Space (idCoordinateSpaces.idPasteboardCoordinates,
Rem idCoordinateSpaces.idParentCoordinates, or
idCoordinateSpaces.idInnerCoordinates)
Rem The coordinate space to use for the transformation.
Rem
Rem From (as an array of two numbers, an idAnchorPoint enumeration,
Rem or a idBoundingBoxLimitsEnumeration):
Rem The center of transformation.
Rem
Rem With Matrix (Transformation Matrix):
Rem Specify the matrix used to transform the object.
Rem
Rem Replacing Current (Boolean):
Rem If true, replace the current transformation values of the object
Rem with the values of the transformation matrix.
Rem
Rem Considering Ruler Units (Boolean)
Rem If true, use the current ruler usings and zero point location
Rem when transforming the object. If false, use the coordinates
Rem relative to the Coordinate Space parameters.
Rem
Set myInDesign = CreateObject("InDesign.Application.CS3")
Rem Template for TransformationMatrices.Add
Rem Add([HorizontalScaleFactor], [VerticalScaleFactor], [ClockwiseShearAngle],
[CounterclockwiseRotationAngle], [HorizontalTranslation], [VerticalTranslation],
[MatrixValues]) As TransformationMatrix
em Note empty parameters.
Set myRotateMatrix = myInDesign.TransformationMatrices.Add(, , , 27)
Set myScaleMatrix = myInDesign.TransformationMatrices.Add(.5, .5)
Set myShearMatrix = myInDesign.TransformationMatrices.Add(, , 30)
Set myDocument = myInDesign.Documents.Add
Set myRectangle = myDocument.Pages.Item(1).Rectangles.Add
myRectangle.GeometricBounds = Array("72pt", "72pt", "144pt", "288pt")
Rem Template for Rectangle.Transform:
Rem Transform(In As idCoordinateSpaces, From, WithMatrix, [ReplacingCurrent],
[ConsideringRulerUnits As Boolean = False])
yRectangle.Transform idCoordinateSpaces.idPasteboardCoordinates,
idAnchorPoint.idCenterAnchor, myRotateMatrix
Rem Create another rectangle.
Set myRectangle = myDocument.Pages.Item(1).Rectangles.Add
myRectangle.GeometricBounds = Array("72pt", "72pt", "144pt", "288pt")
Rem Transform the rectangle.
myRectangle.Transform idCoordinateSpaces.idPasteboardCoordinates,
idAnchorPoint.idCenterAnchor, myScaleMatrix
Rem Create another rectangle.
Set myRectangle = myDocument.Pages.Item(1).Rectangles.Add
myRectangle.GeometricBounds = Array("72pt", "72pt", "144pt", "288pt")
```

```
Rem Transform the rectangle.
   myRectangle.Transform idCoordinateSpaces.idPasteboardCoordinates,
   idAnchorPoint.idCenterAnchor, myShearMatrix
```

Scripting Documentation Corrections

The following are known errors in the Adobe InDesign CS3 scripting documentation:

- On page 4 of the *Adobe InDesign CS3 Scripting Tutorial*, we refer to ".spt" as a valid AppleScript file extension. This should be ".scpt". Note, in addition, that ".as" is not a standard AppleScript file extension, but is supported by InDesign.
- On page 6 of *Adobe InDesign CS3 Scripting Guide: AppleScript*, the file path cited in the section on compilation is incorrect. The correct file path is:

  ~/Users/*user_name*/Library/Caches/Adobe InDesign/Version 5.0/Scripting Support/4.0

  where "~" is your system volume and *user_name* is your user name.
- On page 6 of the *Adobe InDesign CS3 Scripting Guide: VBScript*, the file path cited in the section on compilation is incorrect. The correct file path is:

  ~:\Documents and Settings\All Users\Application Data\Adobe\InDesign\Version 5.0\Scripting Support\4.0

  where "~" is your system volume.
- On page 6 of *Adobe InDesign CS3 Scripting Guide: JavaScript*, the reference to "InDesign CS_J" in the section on script interpretation should refer to "InDesign CS."
- On page 27 (AppleScript version), 24, (JavaScript version), and 25 (VBScript version) of the *Adobe InDesign CS3 Scripting Guide*, we refer to a web page that has changed. The new link for the XMP specification is:

  http://partners.adobe.com/public/developer/xmp/topic.html

Known Issues Related to InDesign CS3 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 CS3 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 will not create the persistent engine.

Event Listeners Added or Removed During Event Propagation Are Not Handled According to the W3C Specification

The *W3C DOM Level 2 Event Handling 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."

And:

> "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.