

# Chapter Scripts List

Scripts for chapters are designed to demonstrate concepts covered in the chapter. The code is not intended to be complete scripts but to act as a resource. Emphasis is on creating and using handlers as building blocks in developing workflow solutions to repetitive and otherwise time-consuming tasks.

## Chapter 1

**Proofsheet** - Uses active document allowing vertical or horizontal format. Prompts user to choose folder of image files. Assumes that all files in folder chosen are valid image files. User is presented custom dialog in which to set number of rows, columns, and other parameters for the pages. Pages are added to document as needed. Border width only applies if fit option is set to fit frame to content with black, solid border only. If a page item is selected, the selected area will be used to define the image grid, otherwise, the live area of the page will be used.

**Ruled Form** - Creates a ruled form from user input. Custom dialog allows up to 5 columns for form. If a column is given a width of 0, the balance of the form's width will be calculated for this column. If a column is given a width of -1, it is ignored. Assumes Myriad Pro and Minion Pro fonts are installed.

**Ticks** - Same as Tick Marks script with exception that parameters for ticks are hard-coded in the script.

**Tick Marks** - Draws tick marks to the corners of selected page items. Groups ticks for each item selected. Provides user a dialog in which to define tick length, tick offset and tick weight; also if ticks should be placed on geometric or visible bounds.

## Chapter 2

## Chapter 3

**Hello World** - Demonstrates hierarchical nature of InDesign. Creates document with text frame on first page of first spread of document.

**Sample Setup** - Sample script for setting up default preferences for InDesign.

## Chapter 4

**Test Ruler Origin** - Demonstrates problem that can occur when ruler origin is set to spread origin in a facing page document.

**Live Bounds TextFrame** - Creates a text frame based on the live bounds area of a page.

## Chapter 5

**Apply Font\_Every** - Sets font for every word in the active document matching string defined.

**Calculate Grid** - Calculates bounds for items to be created in a grid defined by rows, columns, gutter, and gap. Returns list of lists; does not create the items.

**Check Selection** - Demonstrates working with selection. Gets list of selected items if the list contains an item of the designated class.

**Delete Swatches List** - Deletes the color swatches for the active document ignoring protected colors defined by name in a list.

**Delete Swatches\_Try** - Uses a try statement to delete all non-protected swatches for the active document. Note: Protected swatches will throw an error if a script tries to delete.

**Delete Using Every** - Removes every empty text frame for the active document.

**Name Items** - Names page items of selection that match the class designated. Names are appended with a number that is padded with a leading zero for numbers less than two digits. Will not work in versions prior to CS5. The name to be applied ("Image") is hard-coded in script.

**Make Rectangles** - Creates a number of rectangles defined by list of bounds lists (list of lists). Expects an open document.

**Step and Repeat Rectangles** - Creates grid of rectangles based on bounds list calculated for number of rows and columns with gutter and gap. Values are hard-coded in script. Area defined for grid is defined by page item selected. If no selection, the live area for the page defined by user is used.

## Chapter 6

**Create Document Project** - Demonstrates using a script template (Document Template) to create a similar script. Create Document\_Final is an example of a resulting script.

**Create Guides** - Demonstrates using InDesign's built in Create Guides command. Script creates document using "8x10\_Facing" document preset and creates guides on pages 2 and 3 of document (Layer 1).

**Detach** - Demonstrates detaching an overridden master page item.

**Document From Preset** - Creates a document preset if one as named does not exist. A document is then created using the preset.

**Folio Frames** - Creates footer frames on the master pages of master spread "A-Master." These are then populated with the automatic page number.

**Get Layer Reference** - Returns reference to layer in active document if found. Otherwise, a layer is created and moved to the back if the toBack variable is set to true. Script makes sure that layer, if existing, is not locked. The status of the layer and the index of the active layer before the new layer is created is also returned.

**Master Based On** - Demonstrates creating a master spread based on another.

**Master Page Items** - Creates guide and text frame on page 2 of a master spread.

**Master Pages** - Creates a document from the document preset "8x10\_Facing." Adds a CMYK color using createProcessColors handler. Overrides pages per document and settings for column count on a master spread page.

**Merge Layers** - Merges all layers for the active document into the top layer.

**Overrides text frame on the associated page (page 1).**

**Ruler Guides** - Demonstrates creating a number of ruler guides using a list of positions for both horizontal and vertical guides. Requires facing-page document.

**Spread With Spine** - Demonstrates creating a three-page spread and resizing the center page as with a spine for a book cover.

## Chapter 7

**Check Open Document** - Determines if document named in script is open. If it is open, but not the front-most document, it is brought to the front. If not open, the user is asked to find the file. If the user selects a file other than as designed, the user is given the option of using the file chosen instead.

**Document From Template** - The template saved using the Simple Template script is opened and saved at a location chosen by the user.

**Get File Names** - Uses AppleScript's info for command to determine the file type and name extension for files in a

designated folder. It returns a list of names for files whose file type and name extension match the designated values.

**Load Masters** - Presents user with list of templates from which to choose. The master spreads are loaded to the active document with the global strategy for master page load set to false by default (master spreads will not be renamed in case of name clash).

**Open to Zoom** - Demonstrates getting values for InDesign's layout window and using the values in a script to automatically open documents at the desired size and zoom percentage.

**Read Text File** - Demonstrates reading text from a text file using an optional delimiter. The default value for the delimiter is a return character.

**Read Numbers** - Demonstrates one way to read a file of numbers.

**Save Document** - Checks the active document to see if it has been saved, converted, or modified. If not saved, the user is asked to determine the file path for the document to which it is then saved. If the document has been saved, but since modified, the document is saved.

**Say To File** - An example of creating a file using path to and choose file name. It also demonstrates using AppleScript's say command.

**Simple Template** - Creates a document preset from which a document is created. The document is then saved as a template ("Letter Single") in the Templates folder for the application.

**Write to File** - Demonstrates writing text to file with option to append. Append boolean is set to true by default.

## Chapter 8

**Create Process Colors** - Creates any number of process colors from a list hard-coded in the script. The handler used returns a list of the colors named which were already existing in the document and not changed by the script.

**Create Tints** - Creates a series of tints for the designated swatch for the active document. By default, the script uses the swatch "Black."

**Doc From Chosen Preset (Choose Style Project)** - Uses a script template to create a script that allows user to select a document preset from a list of presets. It then has the user select a style sheet from which to load color swatches. A document is then created using the preset with the swatches loaded.

**Load Swatches** - Loads swatches to the active document from a style sheet named in the script. The style sheet named in the script is "Colors.indd".

**Merge Swatches** - Gets a list of all spot colors found in the existing document and merges with Black.

**Replace Swatches** - Replaces the active document's colors designed by name in a list with a replacement color (replaceColor variable is set by default to "Black").

**Two Stop Gradient** - Demonstrates creating a gradient from two swatches named. The script establishes the mid point for the gradient, although this value is optional.

## Chapter 9

**Chained Frame Style (project)** - Combines the Chained Styles and Create Frame Style scripts into a script that creates a frame style supporting chained paragraph styles.

**Chained Styles** - Demonstrates creating a chain of paragraph styles where each style designates its next style.

**Create Character Styles** - Demonstrates setting up a series of default character styles for the application.

**Create Frame Style** - Demonstrates creating an object style for a text frame.

**Import Styles** - Demonstrates using importStyles handler for importing styles from a single style sheet.

**Span Column** - Demonstrates using the new span/split column capability available in CS5 and above.

## Chapter 10

**Article (Article Project)** - Demonstrates creating a document using a template, importing the styles from a style sheet, flowing text from a designated file into a text frame that is styled using a chained frame style. Metadata is added to the document and the document is saved. User creates alias to the script and runs from InDesign's Scripts panel. This takes advantage of the script's alias link to its parent folder and its resources.

**Break Link** - Breaks the story flow after the frame selected by the user without reflowing the text.

**Doc With Linked Text (Linked Text Project)** - Creates a document with linked text. It is part of the Linked Text project.

**Fit Linked Frame** - Takes advantage of the vertical balance columns property of a text frame to fit the last text frame of linked frames to its text.

**Name First Frame** - Names the first frame of all linked frames selected. Adds a padded number to the base name which is designated by the user.

**Open Document With Linked Text (Linked Text Project)** - Demonstrates opening a document with linked text without losing text formatting.

**Place File\_Placeable** - Has user select a file to place. Uses InDesign's placeable file types and placeable file extensions to determine if a chosen file can be placed in the active document. If placeable, the file is placed.

**Story Index** - Demonstrates one way to discover the index of the first story of a document not part of a master spread.

## Chapter 11

**Color Image** - Demonstrates how a fake duotone can be created from a black and white or grayscale image in InDesign.

**FPO (For Placement Only)** - Uses relinking to switch low-resolution images to corresponding high-resolution images.

**Place Gun** - Demonstrates getting references to a list of files in a folder and using the list to load InDesign's place gun.

**Place Image** - Has user select file. Checks file type and file extension for placeability using InDesign's placeable file types and placeable file extensions before attempting to place.

**Place Multipage PDF** - Demonstrates how a multiple page PDF can be placed into an InDesign document.

**Place Named Images (Place with Caption Project)** - Involves two scripts: Image Template and Place Images\_Named. Image Template creates a layout with containers named to correspond to names of image files that will later be placed into the document.

**Place With Caption (Project)** - Places the file referenced and uses its file metadata to add a credit and/or caption. Sample document provides object styles for photo and caption frames, and text styles.

**Show (Project)** - Opens file defined by variable theSource. If status of any link is link missing, the script shows the missing graphic to the user. Expects document at pre-defined location: [User folder]:AppleScripts for CS5:Chapter 11:Show:ShowMissing.indd. Save this script to InDesign's Scripts panel before running.

**Update Links** - Updates links that are out of date for the document referenced. Returns number of links updated.

## Chapter 12

**Fence It** - Draws a border around the combined geometric bounds of page items selected. It does this by temporarily grouping the items, acquiring the geometric bounds of the group, and then ungrouping.

**Fence It\_No Group** - Creates a border around the combined geometric bounds of page items selected. It does this by comparing the maximum and minimum bounds of the selection with each of the items.

**Get Grouped Style** - Verifies if a style within a style group exists.

**Group Style By Name** - Uses the all object styles property for the application to verify if a line style exists and the style group to which it exists.

**Line Object Style** - Demonstrates creating a line object style within a style group named “Lines.”

**Rounded Corner** - Demonstrates using object styles to get around the bottom right corner problem in CS5 and CS5.5.

**Rounded Rectangles** - Demonstrates using convert shape to create page items with uniform rounded corners. Only supports the rounded, beveled, and inverse corner style options.

**Special Line Styles** - Creates a display showing special line styles added to the application.

**Stroke Corner Adjust** - Demonstrates how various settings for stroke corner adjustment can affect the length of dashes and gaps.

**Using End Caps** - Demonstrates how rounded end caps can be used for a line to produce lines with a specialized dot pattern.

## Chapter 13

**Based On Cell Styles** - Demonstrates setting up cell styles for a table where the majority of the properties are inherited from a style on which the styles are based.

**Box Text (Project)** - Creates a single-column table to draw a box around text selected in a story flow.

**Calendar (Calendar Project)** - Creates a 12-month calendar using tables populated with a text list (the days of the month). Uses a template with containers named for the month name, the images, and the calendar table. Includes a JavaScript do script that creates a non-modal dialog to indicate the script is processing.

**ColorHead Table Style (Table and Cell Styles Folder)** - Creates a table style where styling is controlled by table properties. Requires selected insertion point.

**Convert Text to Table** - Demonstrates creating a table by converting designated text to a table.

**GrayHead Table Style (Table and Cell Styles Folder)** - Creates a table style with styling determined by table regions which have applied cell styles. Use script with document modified using Based on Cell Styles script. Creates style in active document but not table.

**Image With Caption\_Table (Image With Caption Project)** - Uses a single-column table to contain a graphic file and its caption.

**Import Table Styles** - Uses the importStyles handler to import table and cell styles from a designated style sheet.

**Inline Sidebar (Project)** - Creates sidebar text using a table.

**Read File for Table** - Reads text from a file into a list for populating a table. As part of reading the file, it gets the length of the delimited text to determine the number of rows and columns required for the table. Select “TabReturn for Table.txt” when prompted.

**Total Table Columns** - Requires that a frame for the table named “TableFrame” exists for the active document. The table is created and populated with numbers which are then totaled for each column.

## Chapter 14

**Apply Character Style** - Demonstrates applying a character style to a range of text.

**Create Test Document** - Creates a text frame with a paragraph style named "Text\_14." A live bounds text frame is populated with placeholder text for testing.

**Delete Empty Paragraphs** - Deletes empty paragraphs using a looping mechanism.

**Folio** - Modifies the Folio Frames script to create folios with automatic page number special character..

**Fractions\_OTF (Project)** - Requires text frame in active document with Adobe Caslon Pro type font. Takes advantage of the fractions feature of an OpenType font to style fractions. For this, metacharacters are used for find/change.

**Ornamental Character** - Demonstrates creating a design with ornamental characters found in an OpenType font.

**Random Snowflakes** - Uses ASCII numbers for a special text font to create a random pattern of special characters on a page.

**Style Paragraphs\_Dlg (Style Paragraphs Project)** - Presents the user with a dialog from which to select a paragraph style for each paragraph in a selected set. This is then applied to all paragraph sets having the same number of paragraphs. Use with "MakeDescPrice.indd" document and follow instructions in the document.

**Style Paragraphs\_Loop (Style Paragraphs Project)** - A handy mathematical formula takes advantage of AppleScript's mod function to style a repetitive paragraph style pattern.

**Test Story** - Test script for experimenting with references to text.

## Chapter 15

**Capitalization** - Makes document with paragraph style and applies the style to text.

**Drop Cap\_Charstyle** - Demonstrates creating a character style and using the character style as part of the properties for a drop cap. Assumes paragraph selected in document with swatch "C=100 M=0 Y=0 K=0."

**Get Bullet Properties** - Demonstrates getting the properties from a bullet paragraph style and placing in a record to be used in defining the style in a script.

**Grep Style\_Ordinals (Formal Invitation project)** - Styles all ordinals that follow a number to a character style for which OTF ordinals have been set to true.

**Nested Line Style (Car Ad Project)** - Creates paragraph and character styles for nested style. Using named text frames, the text is styled using the nested line style.

**Reverse Text** - Demonstrates creating reverse text using a rule above.

**Rule Below** - Demonstrates setting rule below values for a paragraph.

**Sentence Case** - Changes capitalization of selected text to sentence case.

**Style Markup** - Creates a paragraph style that applies a character style to every character string included in angle brackets to which it is applied.

**Style Program (Program Project)** - Demonstrates creating a nested paragraph style to set the styling for a typical program.

**Underline and Strike Through** - Demonstrates similarities and differences of underline and strike through.

## Chapter 16

**Find Change By Attribute** - Versatile find change script that allows literal find, find/change, and attribute find, and find/change.

**Find by Charstyle** - Returns reference to text in object referenced that is styled with a designated character style.

**Find Change Grep\_Attrib** - Demonstrates using a script to find and change text defined using a GREP expression.

**Find Change Text** - Finds and changes literal text within a text selection.

**Find Text** - Returns reference to text found that matches the text string identified.

**Fractionator (Fractions project)** - Uses GREP expressions to find and change number-slash-number GREP expressions to a “real” fraction.

**Glyph\_Apply Style** - Demonstrates how you can apply character styling to a designated glyph.

**Metacharacters (Project)** - Uses metacharacter string for find/change operation to style prices in ad.

**Recipe (Project)** - In addition to setting fractions using the OTF Fractions feature, it shows how a range of text can be converted to a table to provide a background color for the text.

**Write Found Text** - Searches text for text styled with a designated character style and writes the contents of the text references to a file.

## Chapter 17

**Dialog With Choices** - Creates custom dialog with measurement combobox, a checkbox control, and a radio button group.

**Dialog With Columns** - Creates custom dialog using columns to align widgets vertically.

**Dialog With Enable Group** - Creates custom dialog with enabling groups.

**Dialog With Measure** - Creates custom dialog with measurement editbox designated for receiving measurements in inches.

**Dialog With Row** - Creates custom dialog with all widgets contained in a single row.

**Dialog With Panels** - Demonstrates creating a custom dialog with border panels used to organize input fields.

**Floating Window** - Calls the first handler to show a non-modal dialog, and then the second handler to hide the window. Also demonstrates creating a document without a window and then creating the window once processing is complete.

**Step and Repeat Rect\_Dlg** - Adds custom dialog box to Step and Repeat Rectangles script from Chapter 5.

## Chapter 18

**Content Type** - Creates rectangle and assigns its content type to graphic type.

**Convert Shape** - Creates a rectangle on page 1 of existing document, then duplicates the shape. It then moves the duplicate and converts it into a triangle.

**Curve Points** - Demonstrates creating a path which has both corner and curved path points.

**Drop Shadow** - Creates a rectangle with a drop shadow. Assumes a document exists with measurements set to points and swatch “C=0 M=0 Y=100 K=0.”

**Duplicate With Transform** - Duplicates a selected path or page item with transformation. The transformation is repeated

within a loop using transform again.

**Entire Path** - Demonstrates getting the path points for a path drawn and selected in a document.

**Get Property Record (Transparency Settings Project)** - Gets transparency fill values of selected item as part of creating a property record. Uses this record to define transparency settings as part of creating a rectangle.

**Gradient Fill Flower** - Uses transformation to create a decorative page element with a gradient fill. Demonstrates adding and modifying path points, using a gradient, and transformation.

**Odd Rule Fill** - Demonstrates how the zero winding rule for InDesign can be used to create an odd rule fill for a path that intersects itself.

**Oval From Center** - Demonstrates creating a page item using its center coordinates.

**Polygons\_5to9 Sides** - Demonstrates creating a series of polygons where the number of sides and center coordinates change within a repeat loop.

**Recreate Triangle** - Demonstrates using path returned from Entire Path script to recreate a triangle.

**Rounded Rectangle** - Uses convert shape to provide another option for creating a rectangle with uniform rounded corners.

**Spirograph** - A fun project that lets you experiment with values for transforming an oval using a custom dialog.

**Test Pathfinder** - Test script provided for testing the various pathfinder operations.

**Test Transform (Project)** - Two scripts to help you understand transformation and transform again.

**Text on a Path** - Demonstrates creating text on a path with a script. Requires a path selection.

**Transform Master Overlay (Master Transform Project)** - Transforms the window overlay of a master page for a page that itself has been transformed.

**Transformation Matrix** - Demonstrates creating a transformation matrix and applying it to a rectangle created in the active document.

**Transform Item** - Creates transformation matrix and uses it to transform an item around a transformation point.

**Transform Page (Master Transform Project)** - Uses a transformation matrix to transform a page.

**Transparency** - Demonstrates creating objects where one has transparency set. Assumes active document with measurements set to points and swatches “C=0 M=0 Y=100 K=0”, “C=100 M=0 Y=0 K=0”, and “Black.”

**Transparency Settings (Project)** - Work through creating property record for getting and setting transparency.

## Chapter 19

**Asset Info** - Returns a report of name and date for assets in open libraries.

**Asset Styles** - Demonstrates how an asset can be used to add styles to a document. If asset is available, it is placed into the active document. Its styles are then used to style an imported text file.

**Choose Library** - Allows user to choose a library from a list of files found in the application's Presets:Libraries folder. If the file does not exist, it is opened.

**Concentric Circle** - Creates a library asset for the library “MyLibrary.indl”.

**Create Library** - Returns reference to library referenced by name. If the library is not opened it is opened. If it does not exist, the user is given the option of creating a new library.



**Custom Text Variable** - Demonstrates creating a custom text variable.

**Extract Label** - Extracts the label named “script” stored in the page item named “scriptItem”. It then runs the script using do script.

**Get Library** - Returns reference to a library referenced by name. If the library is not open it is opened from the application’s Presets:Libraries folder.

**Insert Text Variable** - Script for inserting a custom text variable at point of text selection.

**Label With Arguments (Do Script Project)** - Creates a script in the label of a text frame named “textItem” designed to act on arguments passed to it when it is run.

**Newsletter Final (Newsletter project)** - Adds text variables and metadata to a script template named “Newsletter.” The footers for the document are changed to use a text variable for the publication date. AppleScript’s data type is used to designate special characters for the footer string. The document created with the script will serve as a template for later use.

**Place Asset to Doc** - Places asset named to document and optionally moves it to a page and place point designated.

**Place Asset to Text** - Places asset as an anchored object to text selection. Uses information stored in label to place the asset outside of the selected item’s parent text frame.

**Run Script With Arguments (Do Script project)**- Runs a script saved as a page item label. As part of running the script, arguments are passed to the script. Expects active document with swatch “MyRed.”

**Three Rings** - Demonstrates creating a library asset with a graphic routine that uses transformations. The asset’s size is stored as a custom label in the asset as part of the creation process.

## Chapter 20

**Anchor Placeholder (Business Card project)** - Same as anchor script for Postcard\_Image project with exception that default values are set for Business Card project.

**Anchor Placeholder (Postcard\_Image project)** - Creates anchored image placeholder by placing and anchoring image selected by user to selected text frame. Number values for imageReferencePt, and anchorPt indicate cell of transform reference grid (proxy point). Default values anchor image to top left corner of selected text frame, anchored on the left center transform reference point. Default values are set for Postcard\_Image project. Requires document structured for XML import.

**Anchor Placeholder\_Dlg** - Adds a custom dialog to the Anchor Placeholder script to designate the position of the anchored placeholder in relationship to the selected text frame.

**Anchor to Insert** - Anchors an image placeholder to an insertion point for XML import.

**Aid Attributes** - Illustrates using aid XML attributes to designate paragraph and character styling as part of the XML tag.

**Biz Card\_1up (Project)** - Demonstrates using a Place XML command to place XML in the master text frame for the first page of a document.

**Biz Card 1up\_Multiple (Biz Card\_1up Project)** - Demonstrates how Autoflow can be used to produce a multiple-page document from an XML file having multiple entries.

**Biz Card\_4up (Project)**- Demonstrates an XML import to a document that has repeating records on each page.

**Books (Books project)** - Illustrates using ignore unmatched incoming XML import preference to allow selected XML records to have an extra XML element in their structure.

**Grid Items\_Dlg** - A utility script that can be used to set up a grid of linked master page text frames for multiple record

XML import.

**Map Tags** - Illustrates how tag mapping enables imported XML text to be styled automatically using tag to paragraph style mapping.

**Merge Import (Project)**- Introduces using XML import with merge import set for the import style.

**Postcard\_Multi (Postcard\_noImage Project)** - Uses tagged placeholder text for importing an XML file with multiple records. Tags for fields within the records must match the paragraph styles assigned to the placeholder text.

**Postcard\_Image (Postcard Image Project)** - Places XML to document prepared using Tag Markup Paragraphs and Anchor Placeholder scripts.

**Postcard\_noImage (Project)** - Details how an existing single-page document can be prepared for multiple-page XML import.

**Tag Markup Paragraphs** - Utility script that can be used to automate tagging and marking up text placeholders for XML import.

**XML With Photo** - Illustrates importing an XML file which includes an image tag. Alias of the script needs to be run from InDesign's scripts panel to take advantage of its file path.

**Wines (Project)** - Illustrates using ignore whitespace to import XML to a page structured with text that will be inserted into the XML text flow.

## Chapter 21

**Create Book (Book project)** - Creates a book from files in folder selected.

**Create TOC Style (Book project)** - Imports paragraph styles from a style sheet and creates a table of contents style using the styles.

**Import Styles** - Custom dialog gives user options for selecting style sheet and style format options. Includes option for setting import strategy.

**Index Entry** - Adds an index entry for text selected when the script is run.

**Index Topics** - Gives user the option of reading topics from an existing text file, or importing topics from a standing document such as a style sheet.

**Newsletter Setup** - Uses the Newsletter template (from Chapter 19) to create a newsletter document complete with placeholders for the ads planned, and adds masthead.

**Rule Gutters** - Rules the gutters for a multi-column page.

**Small Ads (Small Ads project)** - Reads column parameters and ad list from a text file. Presents the user with a list of ads planned and fields for entering basic information about the ad: background, border, and text frame. Creates the ad structure and saves it in a folder named for the month designated in a pre-defined location. Values for metadata information will need to be defined in the script.

## Chapter 22

**Edit Original** - Demonstrates using Edit original to open a selected image in Photoshop.

**Image With Caption (Newsletter project)** - Places image chosen by user and adds caption frame with credit and caption information from image's metadata (Author and Description fields). The image and caption frame are grouped and moved to center of page.

**InCopy Setup** - Creates a story structure for InCopy given the planned length for the story. Designed for copy writers who

are working remotely. When the story is written, it then can be opened in InDesign and added to the workflow for the publication. Requires style sheet named “InCopy Basic” to be in the Presets:Styles folder for InCopy.

**Place Linked Table (Microsoft Excel project)** - Places and links a Microsoft Excel spreadsheet.

**Place Microsoft Word\_Style (Microsoft Word project)** - Places a Microsoft Word document in InDesign and, using find/change, converts drop caps and bullets style overrides to character styles “DropCaps” and “Bullets” respectively.

**Save Address Book Data** - Demonstrates pulling information from Address Book into a tab-delimited file ready for data merge. Requires entries in group as named (change name of group as needed) having first name, last name, and address fields. The address field needs to have street, city, state, and zip.

**Test Droplet** - Demonstrates creating a script application that can act as a droplet or application. Example processes files in folder, moving them to folders by file extension.

**Newsletter Project** - Uses a number of scripts for automating a newsletter. Creates document from Newsletter.indd template, creating placeholders for ads using information read from text file, and more.

**Place Ads (Newsletter Project)** - Replaces placeholders in document with completed ads. When this script is run, the user selects the appropriate month for the publication. The month chosen should correspond to the month designated when working with the Small Ads project (Chapter 21).

**Place Gun (Newsletter project)** - Allows user to select a number of images to place. Images layer is activated and default graphic object style is set for Image style to assure that images are placed on the correct layer and assigned the style as part of being placed.

**Story Placeholder\_InCopy and Story Placeholder (Newsletter project)** - Creates story placeholders in the Newsletter document for stories that have been planned. The information for the planned stories is hard-coded in the script.

**Update Excel Table (Microsoft Excel project)** - Demonstrates how a document with a linked Excel spreadsheet can be opened and the spreadsheet updated without losing table styling.

## Chapter 23

**Create Bookmarks (Interactive PDF project)** - Bookmarks document using paragraph style located on pages to be bookmarked. User selects paragraph style for bookmark.

**Export Slideshow (Slideshow project)** - Exports document as PDF slideshow allowing user to select page transitions.

**Hyper Buttons (Interactive PDF project)** - Creates buttons from page items for page navigation. References all of the rectangles on a master page named using a specific naming convention (file name, underscore, page number). Hyperlinks the rectangles to pages designed in the button's name.

**HTML Export** - Gives user a choice of three options: structure document, export to HTML, and both. Structure document creates articles for each page item in the page in the order placed, from first to last. When the file is exported, the article panel order is used. The document is then exported to HTML using predefined export preferences including settings to honor anchor placements.

**Hyper Glossary** - Relies on glossary items in a text frame named “Glossary.” It gets a list of all the glossary entries identified by their assigned character style. It parses this list and, using a find operation, locates matching text in the document set “Body” style. The text found is styled with a character style, and a hyperlink is created to the glossary page.

**Key Frames** - Demonstrates how key frame settings can be part of animation path for precision timing.

**Movie With Buttons** - Places movie with poster file, buttons, and sound file to respond to buttons being clicked. Expects files named in script to be in Movies folder for user.

**Multistate Object (Multistate project)** - Creates a multistate object using image and text files found in corresponding

folders. File names for images and text files are the same with exception of extension.

**Multi Animation (part of Animation project)** - Demonstrates using timing groups and timing settings as part of an animated document.

**Place Movie** - Places movie and poster designated in script. Container for movie is named "Movie." Expects movie and poster files to be in Movies folder for user.

**Prepare Slideshow** - Requires rectangle selected or named "Image" on document's A-Master spread. This is used to populate document with images in "Images" folder for the project. As part of placing the image, it gets the image's metadata (document name, author, and description) and uses it to populate its caption frame.

**Three State Buttons** - Creates buttons next to rectangle named "Movie". Buttons have three states: roll over, clicked, and normal.

## Script Templates

**These scripts are found in the folder for Chapter 17 as they demonstrate using custom dialogs to get the information needed for a script. Although some can be considered to be complete, they are designed to be used as a starting point for scripts requiring a similar custom dialog and/or handlers.**

**Border It** - Uses live bounds of active page to create a bordered rectangle. User interaction dialog provides selection or object style, stroke weight, color, tint, gap color, gap tint, corner radius, inset from bounds (may be negative), and stroke alignment. Handlers: getLiveBounds, set/resetMeasures, calculateBounds, createBorder, getStyleNames, getColorNames, getStrokeNames, getLayerNames, testStyles.

**Dialog Template** - This dialog makes no sense. It contains just about every widget available, so makes a good model for cutting and pasting code into your own script requiraaing a dialog.

**Search Template** - Provides dialog for entering paragraph and/or character style, and option criteria for find.

**Step and Repeat** - Returns values entered by user in custom dialog box. If user selects object style enabling checkbox, values in Set Properties enabling group will be ignored. Handlers: getLiveBounds, set/resetMeasures, calculateBounds, createBorder, getStyleNames, getColorNames, getStrokeNames, getLayerNames, testStyles

**Web Doc Choose Preset** - Document created from document preset. Allows user to choose preset from list of presets that have web intent. Handlers: webDocFromPreset, createLayers, customDialog, getListMatch, removeString\_List.