

101 FORMS eTíps

ted padova

<http://www.west.net/~ted>

<http://www.acrobatusers.com>

<http://www.acrobatusers.com/blogs/tedpadova>



welcome

LICENSE AGREEMENT:

By opening this file you agree to the terms and conditions of this License Agreement.

1. This document is yours! You are free to do with it as you wish without reservation.
2. You can distribute the document to whomever you wish.
3. You can modify the document as you see fit.
4. You can claim any or all information as your own without giving credit to the author (this will be our secret).
5. You can use the Replace Pages command in Acrobat and eliminate any unwanted photos you see. You can replace the photos on this document with your own images thereby acknowledging you as the author of this file.
6. You can cut, copy, and paste any or all information into other documents of your own design.

ted, I understand, and I

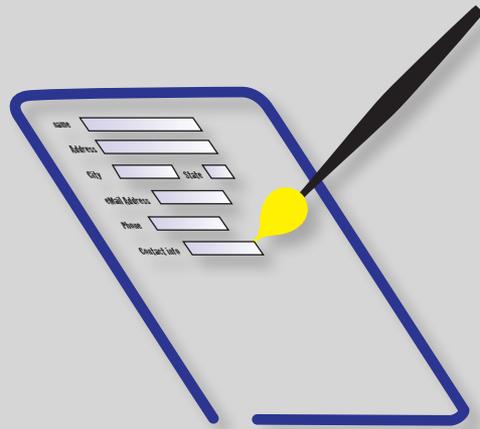
Agree

Do not Agree

Clicking agree and saving the file will eliminate this annoying message.



Designing Forms

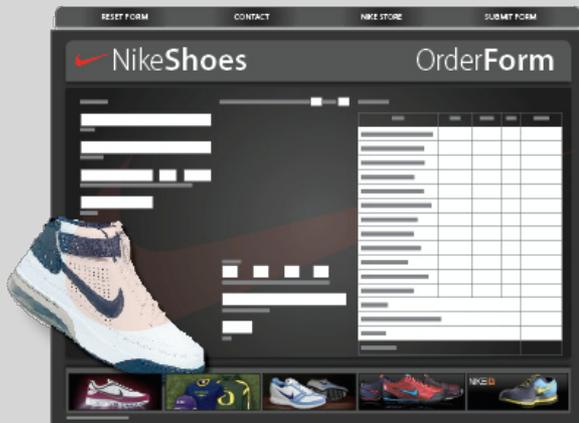
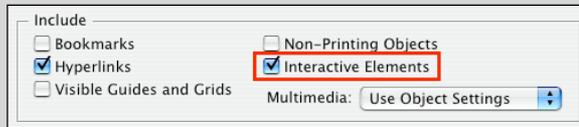


Indicates that the procedure cannot be executed in Adobe Reader

Breaking Page Borders

Graphics and text that fall off a page can only be handled in Adobe InDesign. To design a form with objects breaking page borders, do the following:

- Add an object or text to a Master Page.
- The center point of the object must be within the page border.
- Choose Object | Interactive | Convert to Button.
- Apply the Master Page to a document page.
- Export to PDF —in the Export Adobe PDF dialog box be certain to check the box for *Interactive Elements*.



2

Changing Midpoints on Objects

Suppose you want a graphic to fall completely off a page. Do the following:

- Create a 1-pixel new document in Photoshop on a layer.
- Save the file as a .psd file.
- Place the 1-pixel file in InDesign on a Master Page.
- Add the graphic and place on a Master Page
- Group the graphic and the 1-pixel image.
- Be certain the center point of the grouped objects falls within the page border.
- Follow steps in Tip #1 to export to PDF.

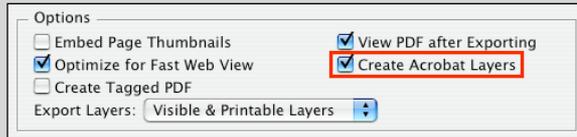


Creating PDFs with Adobe PDF Layers

In order to create Adobe PDF Layers, you need to use an application that supports creating layers. Programs such as Adobe InDesign, Adobe Illustrator, and AutoCAD are among those applications that enable you to export layered files to PDF.

Layered files offer you additional design opportunities for your forms. You might want to change a layer view to play a video file, show a layout for a seating chart or map to a location, or many other similar conditions.

To export to Adobe PDF, be certain to check the box in your application or when using the PDF Maker for exporting layers such as the check box shown when exporting to PDF from Adobe InDesign.



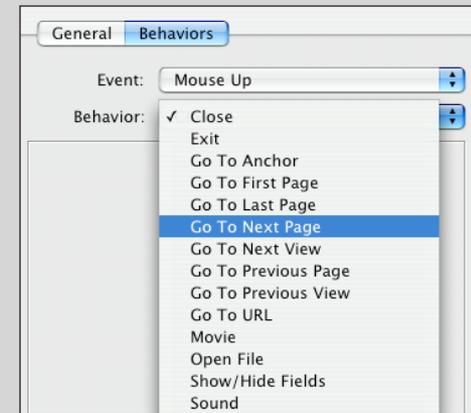
Set the layer view in your application program that you want shown in Acrobat before you export to PDF. The visible and hidden layers in your application document will appear with identical visibility when you open the PDF file.

4 Adding Buttons to Layers

Adding any kind of form field to a PDF in Acrobat applies the field to all layers in the document. If you want to add Button fields that appear only when a given layer is in view, add the buttons in Adobe InDesign. To add a button in Adobe InDesign, do the following:

- ▶ Place an image or add a line of text on a document page.
- ▶ Select the image or text frame.
- ▶ Choose Object | Interactive | Convert to Button (alternately you can open a context menu and choose the same menu command).
- ▶ Click the Select tool and double click an object or choose Object | Interactive | Button Options.
- ▶ Click the Select tool and double click an object or choose Object | Interactive | Button Options.
- ▶ Choose a Behavior from the drop down menu and click the Add button to assign an action to the button.

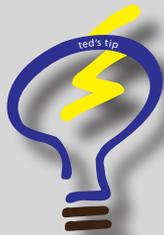
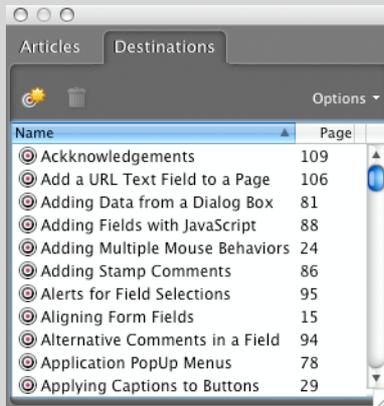
Note that there are only a limited number of actions you can assign to a button in Adobe InDesign. If you don't find the action you want to assign to the button, you can add an Action in Adobe Acrobat.



Adding Hyperlinks in Application Documents

In a word, *don't* add links to views, pages, sections, etc. in an application document such as Adobe InDesign. When you create links and bookmarks in Adobe InDesign and export to PDF, the links are made to *Named Destinations*.

Destinations add a lot of overhead to your file size. When possible, try to avoid creating Named Destinations in your PDF files.



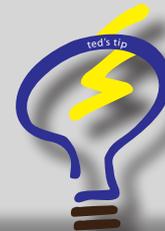
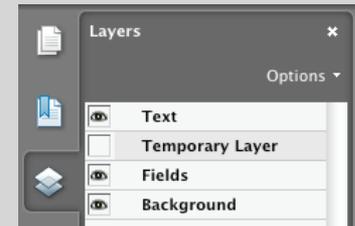
Likewise, don't create links for URLs and email addresses. Acrobat viewers 7 and above are intelligent and automatically recognize URL and email text without creating links. Like Destinations, adding links increase your file size.

6 Designing for RFFR

Run Form Field Recognition works best on forms that have lines identifying field locations. If you create form designs in programs that support layers and your field objects are rectangles, filled boxes with colors or gradients, or other complex design elements, using RFFR won't properly create the field objects.

To overcome the problem, do the following:

- Create a temporary layer in your form design (remember you need to be working in a program that supports layers and exporting to PDF with Adobe PDF Layers.
- Copy the field objects to a temporary layer. If you use boxes for the final design, press the Option/Alt key and drag selected objects in the layers panel to the temporary layer.
- In InDesign or Adobe Illustrator, delete the top, left, and right side of the boxes leaving the bottom line on each box.
- Export to PDF with Adobe PDF Layers.
- Open the Layers panel and select the temporary layer.
- Choose Forms | Run Form Field Recognition.
- Hide the temporary layer.
- In the Layers panel open the Options menu and choose Flatten Layers.



To duplicate selected objects on a layer, open the Layers panel and notice the small rectangle appearing beside the Layer name. Press Option/Alt and drag the rectangle to a new layer to duplicate the selected items.



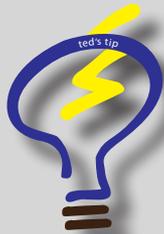
Replacing Pages

The command I use most when creating forms and slide presentations is the Document | Replace Pages command. I always find myself tweaking the page design after I've added fields on a form.

Because Acrobat doesn't provide you a very good editing environment in terms of editing text and images, the ideal method for editing documents is to return to the authoring program, make your changes, and export to PDF.

If you added a number of fields to an original design, then you'll want to change the background data while retaining the field objects. To do so:

- Modify a form in the original authoring program.
- Export to PDF.
- Open the file containing field objects.
- Choose Document | Replace Pages.
- Select the new design and click OK.



I find that when I edit a design and export to PDF I name the new file with a number preceding the file name —something like *0.myFormDesign.pdf*. When it comes time to replace the page, there's no question about what file I need to use for the page replacement

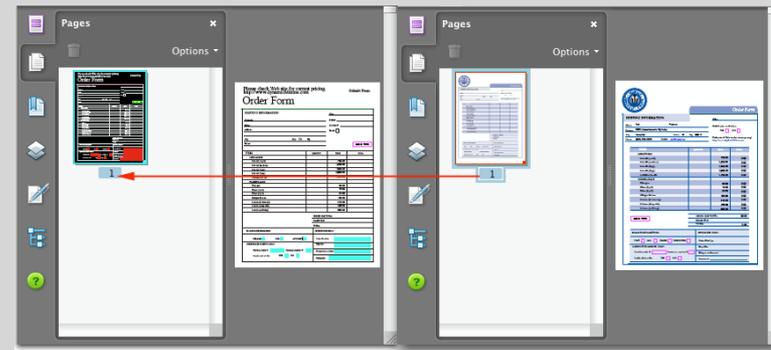
Replacing Pages in Tiled Views



If you have a number of pages to replace in a document or you want a more visual approach to replacing pages, you can drag and drop pages between documents.

Here's how to do it:

- Open two documents in Acrobat. One file is your original design and the other file is the new design.
- Choose Window | Tile | Vertically.
- Open the Pages panel on both documents.
- Click and drag the thumbnail on the new design to the old design containing field objects.
- Move the cursor to the page number below the thumbnail and wait until the page appears reversed in color (black).
- Release the mouse button and the page is replaced.

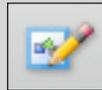


Editing Text on a Form

The ideal, of course, is to return to the authoring program, edit your document, and export to PDF. Next, use the Replace Pages methods described in Tips #7 & 8.

There may be times however where you need to edit a legacy form and you don't have the original authoring application document. To edit text on such a form, do the following (note that this tip requires you to have Adobe Illustrator installed on your computer).

- Click a text block with the TouchUp Object tool.
- Open a context menu and choose Edit Object.



By default, the text opens in Adobe Illustrator. When you select the text you'll notice the lines broken and the paragraph formatting is lost.



- Open the Rulers and drag guidelines to the bounding area of the text block.
- Select all the text.
- Choose Edit | Cut (Command/CTRL + X).
- Click the Type tool in the Toolbox and draw a rectangle to the guidelines.
- Choose Edit | Paste.



- The text is reformed into a contiguous block and can now be edited with word wrap and paragraph controls. Edit the text as needed.



- Choose File | Save.
Be certain to use Save and not Save As.
- View the PDF File.
The text is dynamically updated in the PDF.



Note that if you don't have the same fonts installed on your system that were used in the original form, opening the file in Illustrator will cause font substitution.

If you must edit the form rather than recreating it, choose a typeface close to the original font that you have installed on your system.

Editing Images on a Form

Again, if you have a legacy form and don't have the original application authoring document you can edit images in a program like Photoshop or Photoshop Elements.

To edit an image do the following:

- Click an image with the TouchUp Object tool.
- Press the Option/Alt key and double click the image.
- Edit the image and click Save (not Save As).



Your edits are dynamically changed in the PDF document.

Adding Document Properties

Some authoring applications offer you options for adding Document Property information such as Title, Subject, Author, and Keywords.

If you redesign a form and create several PDFs during the process, you're best off adding Document Property information in the authoring application so you don't have to keep editing the properties on each new file in Acrobat.

- In a program like Adobe InDesign, choose File | File Info.
- Fill in the fields for Document Title, Author, Description, and Keywords.
- Export to PDF

The Document Properties are retained. This information is particularly important when you want to search for forms on your hard drive or server.

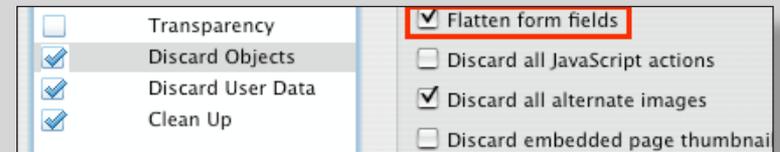
12

Flattening Fields

Field objects like destinations and links add overhead to your files and increase file size. Additionally, form fields add to the file size. If you have a form that no longer requires editing the field data and you want to display the form while retaining the data you can flatten the fields.

Follow these steps to retain data while eliminating field objects:

- Choose Advanced | PDF Optimizer
- Click Discard Objects in the left pane in the PDF Optimizer.

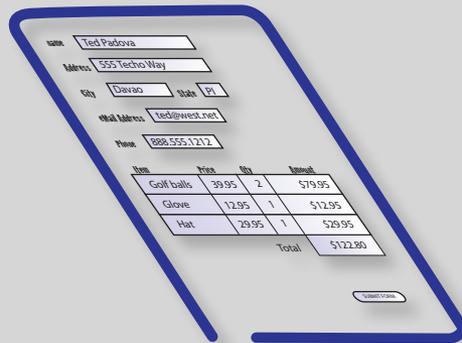


- Click OK.
- Name the file.
- Click Save.



Be certain to provide a new name when prompted by the PDF Optimizer before you save the file. Keep the original file just in case you need to go back and make some edits on the form.

Working with Fields



name Ted Padova
address 555 Techo Way
city Davaco state PT
email address ted@west.net
phone 888.555.1212

Item	Price	Qty	Amount
Golf balls	39.95	2	\$79.95
Glove	12.95	1	\$12.95
Hat	29.95	1	\$29.95
Total			\$122.85



Indicates that the task cannot be performed in Adobe Reader



Using Run Form Field Recognition

RFFR is fine for simple forms not requiring a lot JavaScript and calculations. If you have complex tables and you need to use hierarchical names for your fields to simplify calculations, you're better off not using RFFR.

For simple forms do the following:

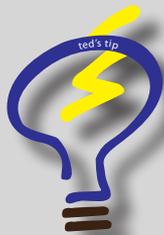
- Open the form in Acrobat Pro.
- Choose Forms | Run Form Field Recognition.
- Edit fields as need to finish up the form.



Scanning for Run Form Field Recognition

You can scan paper forms and use Run Form Field Recognition by following the steps below:

- Scan a form using a high interpolated resolution of 1200 ppi or above.
- Choose Document | OCR Text Recognition | Recognize Text Using OCR.
- Choose Forms | Run Form Field Recognition.



When you use Recognize Text Using OCR Acrobat will straighten a scan within 14 degrees of rotation.



15

Creating Comb Fields

Comb fields are often found on government forms where a user is expected to fill in separate boxes (or divisions with vertical lines) as a single field object.

Comb fields properties are applied to text fields.

To create a comb field, do the following:

- Create a text field.
- Open the Text Field Properties dialog box (double click the field with either the Select Object tool or the Text Field tool or open a context menu on a selected field and choose Properties to open the Text Field Properties dialog box).
- Click Options.
- Remove the check marks for Scroll Long Text and Check Spelling.
- Click the *Comb of* check box.
- Type a value in the characters field box for the number of characters you want to use in the comb field.

Multi-line

Scroll long text

Allow Rich Text Formatting

Limit of characters

Password

Field is used for file selection

Check spelling

Comb of characters



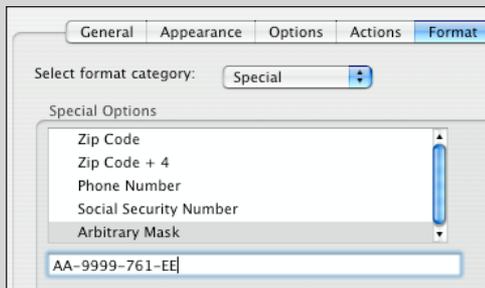
Creating Arbitrary Masks

An arbitrary mask might be used for a section number, an account number, an identification code or similar type of field. The mask can contain alpha, numeric, or both types of characters.

Quite often, you might use an arbitrary mask for an area on a form that might be used for *Official Use Only*. If the user needs to fill in a field with an arbitrary mask, it's helpful to add an instruction or [dialog box](#) informing the user exactly how to fill in the field.

To create a field with an arbitrary mask do the following:

- Create a text field.
- Open the Text Field Properties dialog box.
- Click the Format tab.
- Choose Special from the Select format category drop down menu.
- Scroll the Special Options list and choose Arbitrary Mask.
- Type a code in the field box below the Special Options list.



Note: wildcard alpha characters are A, wildcard numeric characters are 9. Any other character retains the character in the code.

Arbitrary mask field:

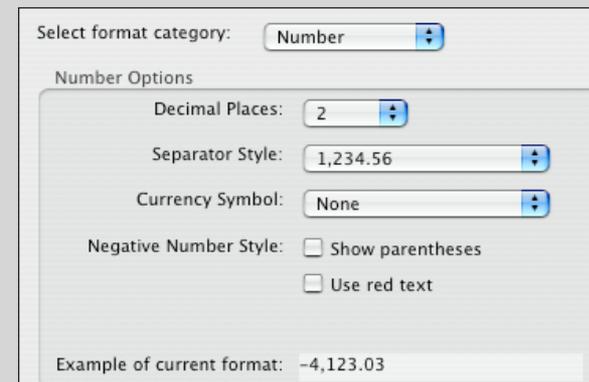
17

Formatting Numbers

If you add calculations to fields you must format the fields as numbers. Before writing JavaScripts, using Simplified Field Notation, or using the formula presets, be certain to format all the fields containing numeric values.

To format text fields as number containers do the following:

- Create a text field.
- Open the Text Field Properties dialog box (double click the field with either the Select Object tool or the Text Field tool or open a context menu on a selected field and choose Properties to open the Text Field Properties dialog box).
- Click the Format tab.
- Choose *Number* from the Select format category menu.
- Select options for the number attributes.



Adding Button Faces

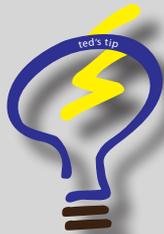
A button face is an icon or an image appearing within a Button field. You can use button faces for importing images or using icons for buttons that invoke actions such as resetting a form, submitting a form, navigating pages, playing and stopping video clips, and a number of other action items you want to assign to a button.

To add a button face to a Button field, do the following:

- Create a button field.
- Open the Button Field Properties dialog box.
- Click the Options tab.
- Choose Icon only from the Layout drop down menu.
- Click Choose icon.
- Click Browse in the Select Icon dialog box.
- Navigate folders on your hard drive to locate a file to import as a button face.

Note that from the Files of type (Windows) or Show (Macintosh) drop down menu you can choose from a number of different file formats to import for your button face.

- Select the file to import and click Select.
- Click OK in the Select Icon dialog box.



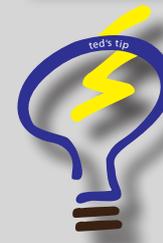
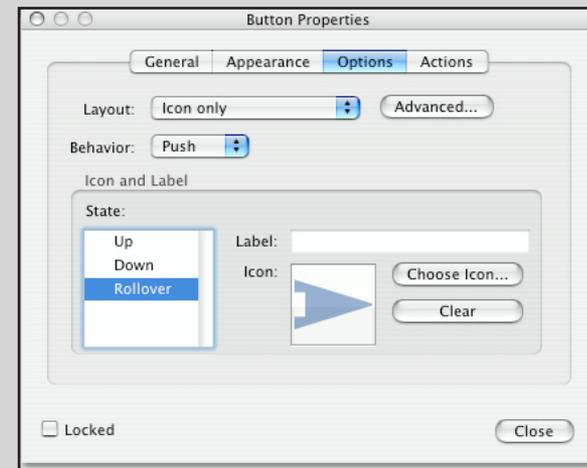
When adding buttons, choose from the options in the Behavior drop down menu. If you choose *None*, you can eliminate the focus rectangle when the button is clicked.

19

Adding Rollovers to Button Faces

To format text fields as number containers do the following:

- Follow steps in Tip # 18 to create a button face.
- Choose Push for the Behavior
- Click Rollover in the State list.
- Follow the same steps for importing a button face.



Creating buttons with transparency is quite simple. For vector objects use Adobe Illustrator. For images, add an image to a transparent layer and save the file as a PDF. All transparency is preserved with both Illustrator and Photoshop layered files.



Duplicating Buttons Across Multiple Pages

A common use for a Button with a button face is navigating pages. If your files are displayed in Full Screen mode, you may want to add navigational buttons to help the end user move around your document.

To create a navigational button and duplicate it across multiple pages, do the following:

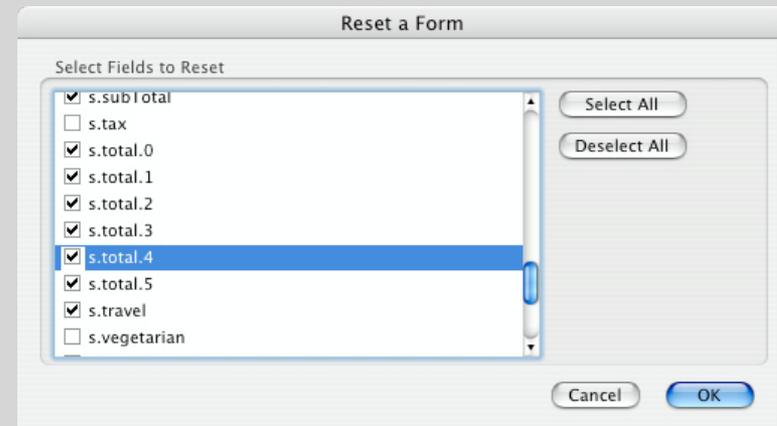
- Add a button to the second page of a multi-page document.
- Click the Action tab and add an action to move to the next page. You can use either an Execute Menu Item action or a JavaScript Action to move to the next page.
- Choose Execute a Menu Item from the Select Action drop down menu and click Add.
- Click *View>Go To>Next Page* in the Menu Item dialog box.
- Repeat the same steps for a button to navigate to the previous page. Click *View>Go To>Previous Page* in the Menu Item dialog box.
- Click OK and Close the Button Field Properties dialog box.
- Select both buttons.
- Open a context menu on the button and choose Duplicate.
- Enter the Page Range in the Duplicate Field dialog box (3 to [last page number]).
- Click OK.
- Click the previous button and choose Duplicate.
- Enter 2 for the Page Range. (Note: if you want a button on page 1 for navigating to the next page, start on page 2 instead of page 3 and follow the same steps.

21

Adding a Reset Button

To add a button to reset the form fields, do the following:

- Create a Button field.
- Click the Actions tab.
- Choose *Reset a form* in the Select Action menu.
- Click Add.
- Check the fields you want to reset in the Reset a Form dialog.



A much better way to reset a form is to use a JavaScript. You have a much easier way to identify fields to clear data if you name the fields properly. See Tip # 59 for more on resetting forms with JavaScripts.



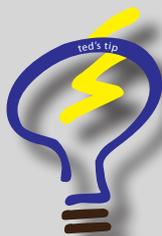
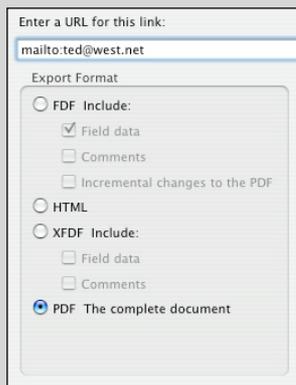
Submitting Forms to eMail Addresses

To create a Submit Form button, do the following:

- Add a button on a form.
- Open the Button Field Properties dialog box and click the Actions tab.
- Choose Submit a form from the Select Action drop down menu.
- Click Add.
- Type an eMail address in the top field in the Submit Form Selections dialog box. The syntax for the line of type is:

mailto:<your email address>

- Check an export format and click OK.



If you want the PDF returned to you by Adobe Reader users, be certain to enable the form with Adobe Reader usage rights before sending your form to Reader users.

23

Creating a Table

One thing to remember when creating tables in Acrobat is to avoid using hierarchical names. In most cases, you'll want Acrobat to rename the first row fields with parent/child names.

To create a table, do the following.

- Add text fields across a row or column and format the fields—especially the number fields.
- Select the fields and open a context menu.
- Choose Place Multiple Fields.
- Enter the number of rows and columns you want for the table in the Create Multiple Copies of Fields dialog box and click OK.

Item	Price	Qty	Amount
Total			



To learn how to straighten up the rows and columns, see Tip # 24.



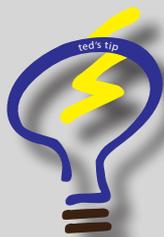
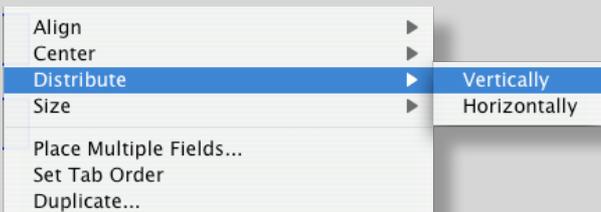
Managing Fields

When you create tables as shown in Tip #23, the fields may not fall within the table area of your design. With a few short steps using context menus you can align fields within any table.

- ▶ Create a table following steps in Tip #23. Don't bother trying to mess around with positioning fields in the Create Multiple Copies of Fields dialog box.

Note that the first row (or column) of fields should be aligned properly before you create the table.

- ▶ Select the last row (or column) of fields and move them to position in the last row (or column) of your table design.
- ▶ Use the Select object tool and drag through the first column (or row) to select only one column (or row) of fields.
- ▶ Open a context menu and choose Distribute | Vertically (or Horizontally).
- ▶ Select the next row (or column) and perform the same steps.



Additional context menu commands appearing when you open the menu on selected fields include aligning fields and sizing them.

Setting Appearance Defaults

25

When creating Button fields I personally dislike the default gray fill on Button fields. If you want to change the default appearance of buttons or any other kind of field, do the following:

- ▶ Add a field to a document page and adjust the Appearance settings as you like in the Appearance tab of the field's properties dialog box.
- ▶ Click Close and open a context menu on the field.
- ▶ Choose Use Current Properties as New Defaults.

When you quit Acrobat and relaunch the program, the new defaults are still honored in Acrobat.



Duplicating Fields

26

For a quick method for duplicating fields, do the following:

- ▶ Create one or more fields on a page.
- ▶ Select the field(s) with the Select Object tool.
- ▶ Press the Option/CTRL key and drag the field(s) to duplicate.



Another quick method for duplicating fields is to select the fields you want to duplicate and press the Option/CTRL key. With the fields selected and the modifier key depressed strike the Right or Left Arrow key.



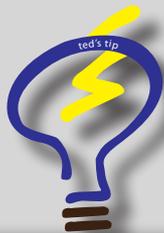
Adding an Address Block

Adobe LiveCycle Designer has a nifty group of fields in the Library panel called an Address Block. See Tip #87 for more on using the Address Block in Adobe LiveCycle Designer.

Unfortunately, Acrobat doesn't have an easy way to add these commonly used fields. Here's a little workaround to help out.

- Create a blank new page by choosing File | Create PDF | From Blank Page.
- Add Fields on the blank new page for last name, first name, address 1, address 2, city, state, zip, email, and phone contacts (i.e., land, mobile, and fax).
- Name the fields xxx.last.name, xxx.first.name, xxx.address1, xxx.address2, xxx.city, etc. The important thing is to use parent/child names and use something like xxx for the root name. (See Tip #28 for instructions on changing field names).
- Save the file to a location where you want to keep templates and commonly used files.
- While the file is still open in Acrobat, choose File | Organizer | Add to a Collection. (Note you must first save your document before the menu command will be active).
- Click a Collection in the Add to Collection dialog box or click the New Collection button to create a new collection and save your file to that collection.

Each time you want to add an address block, open the file from the Organizer and copy fields from the template document and paste the address block in your new form design.



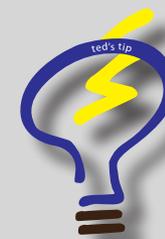
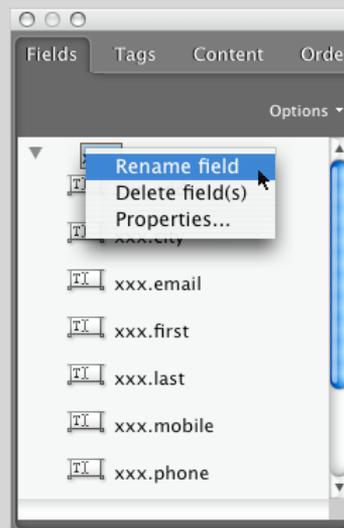
When you create a blank new page you get an annoying text block on the new page. To eliminate the text block and use the blank page for a template, choose File | Save and save the file. When you reopen the file, the text block is gone.

28

Renaming Fields

Take a look at Tip #27 and you'll find an example of one good reason why we would want to change field names. Changing field names that use a hierarchical structure is quite easy in Acrobat. Let's take Tip #27 as an example and do the following:

- Choose View | Navigation Panels | Fields.
- Using Tip #27 as an example, you'll find the root name appearing as xxx. Nested below the root name are the fields with child names.
- Open a context menu on the root name (in this example xxx).
- Choose Rename field.
- Type a new parent name (such as *employee*) and press enter. All fields will take on the parent name and you'll see field names like *employee.last.name*, *employee.first.name*, etc.



You can quickly access navigation panels by opening a context menu on a Navigation Panel button and choose the panel you want to display.



Duplicating Check boxes and Radio Buttons

Check box and radio button fields most often are named with the same identical field name, but they are assigned different Export Values in the Options tab.

If you want to create a table containing a series of check boxes or radio buttons you can't use *Place Multiple Fields* described in Tip #23 where I talked about creating tables. For check boxes and radio buttons you need a little workaround.

- ▶ Create one column or row of check boxes or radio buttons using the Place Multiple Fields command explained in Tip #23.

NOTE: be sure to name the fields using a hierarchical name—something like *xxx.item*. Acrobat will name the fields *xxx.item.0*, *xxx.item.1*, *xxx.item.1*, etc. when you create the table.

- ▶ Distribute the fields so they line up as described in Tip #24.
 - ▶ Select the fields using the Select Object tool.
 - ▶ Press Option/CTRL and drag to duplicate the fields as described in Tip #26. Move the duplicated fields to position in the table.
 - ▶ While the new column (or row) of fields is selected, open the Properties dialog box and click Options. Note that you can edit Options for multiple fields.
 - ▶ Change the export value. If you used the default value Yes for the first column (or row), change the value for the duplicated fields to No.
 - ▶ Click Close in the Properties dialog box.
- At this point you should have two columns (or rows) of fields properly formatted and having different export values for the fields in row 1 vs. 2 or column 1 vs. 2.
- ▶ Select both columns (or rows) of fields.





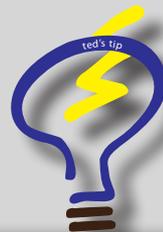
Two columns of checkbox fields

Acrobat User Group Chapters

Location	Group Name	Leader Name	Total Members	Waiting Days	Meet Monthly		Guest Speakers		Workshops	
					YES	NO	YES	NO	YES	NO
location 1	groupName	leaderName	mamb	days 0	m	m				
location 1	groupName	leaderName	mamb	days 1	m	m				
location 1	groupName	leaderName	mamb	days 2	m	m				
location 1	groupName	leaderName	mamb	days 3	m	m				
location 1	groupName	leaderName	mamb	days 4	m	m				
location 1	groupName	leaderName	mamb	days 5	m	m				
location 1	groupName	leaderName	mamb	days 6	m	m				
location 1	groupName	leaderName	mamb	days 7	m	m				
location 1	groupName	leaderName	mamb	days 8	m	m				
location 1	groupName	leaderName	mamb	days 9	m	m				

Acrobat 1
grandT

- ▶ Choose Edit | Copy or press Command/CTRL + C.
- ▶ Open the Fields panel and rename the parent name of the fields as described in Tip #28.
- ▶ Choose Edit | Paste or press Command/CTRL + V. Note that the pasted fields will have different names since you copied them before changing the parent name of the selected fields.
- ▶ Again rename the fields in the Fields panel and move the fields to position. Repeat the steps until you finish adding fields to all columns/rows.



When you paste fields on a form, be certain to have the Select Object tool selected. If the Hand tool is selected and you paste fields, clicking the Select Object tool to move the fields deselects the fields and can make it difficult to reselect them.

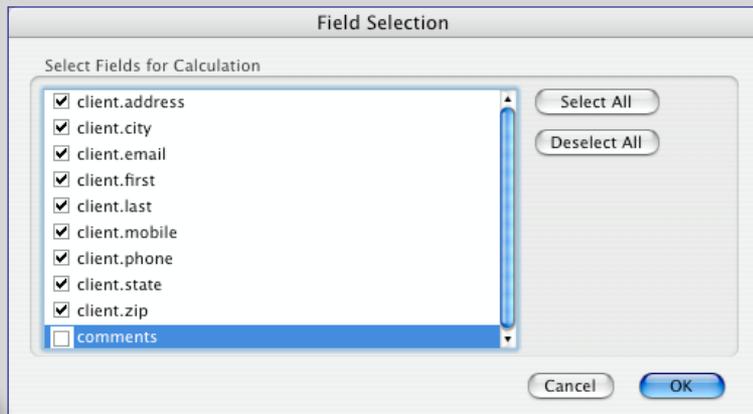
Adding Barcode Fields

You'll want to observe a few precautions when creating barcode fields.

- Populate a form with all the fields you expect to see on the final form.
- Save the form.
- Test the form by filling in all the fields.
- While the fields are populated with data, add a Barcode field (click the Barcode tool and draw a rectangle on your form).

Barcodes need to know how much data are to be added to the code to estimate the size of the field, hence your reason for populating the form with data.

- Click the *Value* tab in the Barcode Properties dialog box.
- Click *Pick* to open the Field Selection dialog box.
- Uncheck boxes for the fields that you don't want to be added to the barcode.



Commercial



**ACROBAT USER
COMMUNITY**

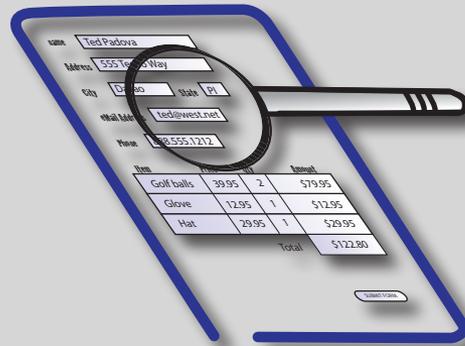


Keep in touch with what's happening with
Acrobat, PDFs, and
LiveCycle Designer on Acrobat Users.

<http://www.acrobatusers.com>



Initial Views



Indicates that the task cannot be performed in Adobe Reader

Setting Initial Views

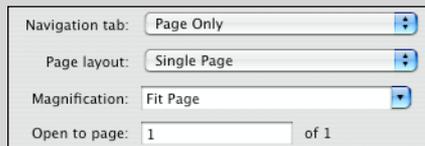
Perhaps one of the most overlooked setting most forms authors make is not setting an initial view for the way they want a form to be displayed on a computer monitor. If left to individual users, preferences for initial views can vary greatly meaning your forms will appear with a different initial view on different users' computers.

The first thing to determine is how do you want your form displayed on a user's system. If you have multiple pages, do you want the Page Layout to be single pages or a continuous page view? What zoom level do you want the user to see when they open your form? Should it open with a 100% view, a Fit in Page view, or a different zoom level?

When you make choices in the Initial View tab of the Document Properties and save your form, the settings prevail on everyone's computer and overrides the individual user's preferences settings.

To set an initial view, do the following:

- ▶ Press Command/CTRL + D to open the Document Properties dialog box.
- ▶ Click the Initial View tab.
- ▶ Choose a Page Layout from the Page Layout drop down menu.
- ▶ Choose a magnification setting from the Magnification drop down menu.
- ▶ Save the form and the Initial View settings are saved with the document.



Navigation tab:	Page Only
Page layout:	Single Page
Magnification:	Fit Page
Open to page:	1 of 1



32

Opening in Full Screen Mode

Full screen mode can be an ideal viewing experience for users filling out a form especially forms that are designed with landscape views.

It's important to add all navigation buttons, print, reset, exit, and submit buttons on forms that you want displayed in Full Screen mode.

To save a file that will open in Full Screen mode do the following:

- ▶ Press Command/CTRL + D to open the Document Properties dialog box.
- ▶ Click the Initial View tab.
- ▶ Check Open in Full Screen mode and save the file.

33

Hiding the Cursor in FS Mode

If you have all the navigation buttons on a form the default cursor that appears when you open a file in Full Screen mode can be distracting for users. With a simple JavaScript you can eliminate the cursor that advances page navigation.

Do the following to hide the cursor when a file opens in FS mode.

- ▶ Choose Advanced | Document Processing | Document JavaScripts.
- ▶ Type a name in the Script Name field box and click Add.
- ▶ Delete the default text and type:


```
app.fs.clickAdvances = false;
```
- ▶ Click OK, click Close in the JavaScript Functions dialog box, and save the file.



Hiding the Navigation Panel

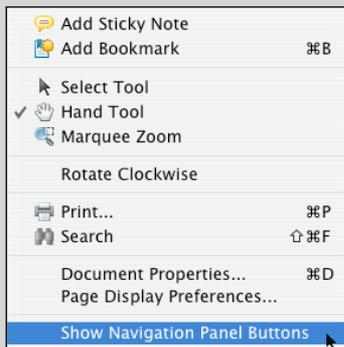
If your forms are intended to be filled in in standard viewing mode you can gain a little more real estate for viewing a form when the Navigation Panel is hidden.

- ▶ Press Command/CTRL + D to open the Document Properties dialog box.
- ▶ Click the Initial View tab.
- ▶ Check *Hide Window Controls* under the User Interface Options.
- ▶ Save the file. The next time you open the document, the Navigation Panel is hidden.

Viewing the Navigation Panel

If you want to temporarily view the Navigation Panel when a file has been saved with Hide Window Controls, do the following:

- ▶ Move the cursor to the far left of the Document Pane.
- ▶ Open a context menu.
- ▶ Choose Show Navigation Panel Buttons.



36

Setting Custom Zoom Levels

When you make adjustments for the Initial View, you choose zoom views from a set of preset magnifications.

If you want to use a zoom view other than a preset, you need to write a JavaScript to set a custom view.

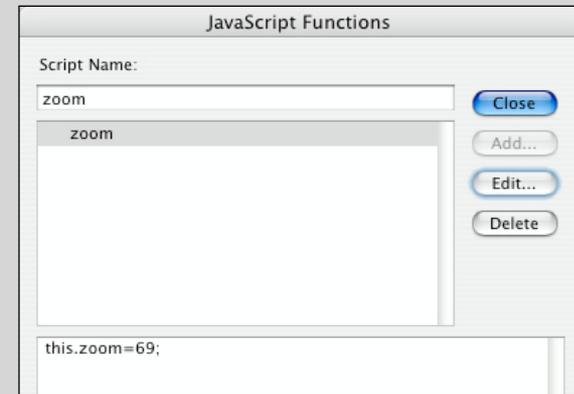
Here's how you do it:

- ▶ Choose Advanced | Document Processing | Document JavaScripts.
- ▶ Type a name in the Script Name field box and click Add.
- ▶ Delete the default text and type:

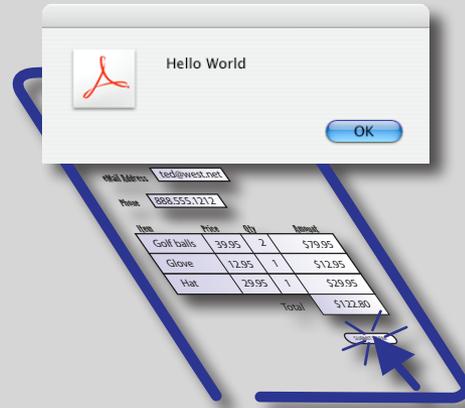
```
this.zoom = 69;
```

The above script sets the zoom view to 69% when the file is opened.

- ▶ Click OK, click Close in the JavaScript Functions dialog box, and save the file.



writing JavaScripts

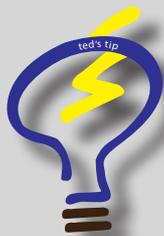


Indicates that the script cannot be executed in Adobe Reader

Navigating Views

On a button action do the following:

- Create a Button field.
- Click the Actions tab.
- Choose *Run a JavaScript* from the Select Action drop down menu.
- Type one of the following scripts in the JavaScript Editor.
 - To go to the next page: `this.pageNum++;`
 - To go to the previous page: `this.pageNum--;`
 - To go to a specific page: `this.pageNum = 0;`
- Click OK in the JavaScript Editor.
- Click Close in the Button Properties dialog box.



Keep in mind that JavaScript is zero based. Therefore if you want to write a script that opens page 7 in a document, your script needs to be written as:

```
this.pageNum = 6;
```

38

Creating an Application Alert

Application alerts can help you save some space on a form. Rather than adding instructions on the form itself, you can create one or more application alert dialog boxes to inform users procedures for filling in a form.

Follow these steps:

- Choose Advanced | Document Processing | Document JavaScripts.
- Type a name in the Script Name field box and click *Add*.
- Delete the default text and type (without the line numbers):
 1. `var msg = "Please fill in all form fields";`
 2. `app.alert(msg);`

Whatever text you add between the quotes in line 1 will appear in the dialog box.
- Click OK, click Close in the JavaScript Functions dialog box, and save the file. Upon saving the file, the dialog box will appear and give you a preview of the application alert.



The above script opens the alert dialog box each time the file is opened because the script is added as a Document JavaScript.

The same script could also be added to a Page Action (open the Properties dialog on a Page in the Pages Panel and click Action and add the script in the same manner), a bookmark, a button, or a Document Action.

Dismissing Application Alerts

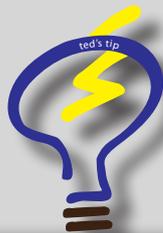
At times you'll want to alert a user when a document is first opened. However, if a user opens your form several times and continually sees the same dialog box appear each time the file opens, it can become quite annoying.

To provide a user an option to dismiss the application alert, use the example code below on a Document Level JavaScript (see Tip #38 for adding a Document Level JavaScript):

```

1.  if(typeof(global.bShowWarning1) ==
    "undefined")
2.  {
3.    global.bShowWarning1 = true;
4.    global.bPersistWarn1 = false;
5.  }
6.  if(global.bShowWarning1)
7.  {
8.    var oCk = {bAfterValue:false};
9.    app.alert({cMsg:"Instructions: Fill out this
    form carefully and click the Submit Form
    button when finished.", nIcon:1,oCheckbox:oCk,
    cTitle:"Reminder"});
10. global.bShowWarning1 = !oCk.bAfterValue;
11. }

```



This button has sample code:

Click Me

40

Adding Line Breaks to Alerts

You might want to format the text a little in an alert dialog box to emphasize a line of type. By adding a few control switches you can easily add line breaks to messages in application alert dialog boxes.

Here's a sample script that adds a line break.

```

1.  var msg = "Here's a script with a line break.\n\
    nVisit my blog at\n\nhttp://www.acrobatusers/
    blogs/tedpadova"
2.  app.alert(msg);

```

In line one the `\n\n` coding creates the line break.

Click Me

41

Invoking Actions on Field Entry

Actions such as application alerts, calculations, and a variety of other action types can be invoked when a user tabs into a field.

To add a JavaScript when a user tabs into a field, do the following:

- Open the field properties on the field where you want the action to be invoked.
- Click the Actions tab.
- Open the Mouse Trigger drop down menu and choose On Focus. This option sets the action to when a user tabs into the field.
- Choose Run a JavaScript from the Select Action drop down menu and click Add.
- Type your script in the JavaScript Editor.

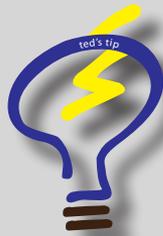
Assessing Viewer Versions

Some features in Acrobat viewers 6, 7, and 8 and above may not be available to users of earlier versions of Acrobat. To help a user understand if a feature won't work in an earlier version, you can have Acrobat assess the viewer and report in an application alert dialog box a message if it detects a version lower than one you want to specify in the code.

To write a script that assesses viewer versions, add the following script to a Document Level JavaScript:

```

1. //is the viewer version less than 8.1
2. if(app.viewerVersion < 8.1)
3. {
4.   var msg = "Some features won't work in
   Acrobat viewer versions less than version
   8.1.";
5.   app.alert(msg);
6. }
```



Line 1 in the above code is a comment line. When you see a line preceded by // remember that the line has no effect on executing the script.

It's a good idea to add comments so that when you want to later review some of your code it will be much clearer to you.

Many of the scripts in this document won't have comments due to space limitations.

43

Assessing Viewer Types

Another assessment function available to you in Acrobat is assessing the viewer type. This code should be added to the forms that have certain actions that can't be executed in Adobe Reader.

To write a script that assesses viewer types, add the following script to a Document Level JavaScript:

```

1. //is Adobe Reader as opposed to Adobe Acrobat
2. if(app.viewerType == "Reader")
3. {
4.   var msg = "Filling out this form requires
   either Adobe Acrobat Pro or Adobe Acrobat
   Standard";
5.   app.alert(msg);
6. }
```



If you want to assess both viewerVersion and viewerType, you can add both scripts to a Document Level JavaScript.

Simply add the above script following line 6 in the viewerVersion script.

Assessing Document Information

You can use JavaScript to report back to you certain information about a document such as Title, Author, Subject, Keywords and more.

To write a script that reports the PDF author on a button action, add the following script on a button:

1. `var f = this.getField("textField");`
2. `f.value = ("Author is: " + this.author);`

Line 1 identifies the text field where the information will be reported. The script is written on a button and when clicked the author of the PDF is shown in the text field.

Coding for other information that can be reported in the JavaScript Console (see Tip #47 for more on using the JavaScript Console) or within a text field includes:

```
this.title; // title field
this.subject; // subject field
this.keywords; // keywords field
this.producer; // PDF producer application
this.creator; // authoring document
this.creationDate; // date PDF was created
this.modDate; // date PDF was modified
this.pageNum; // page number of current page
this.numPages; // number of pages in the PDF
this.filesize; // the file size of the PDF
this.path; // directory path for file location
this.numTemplates; // # of templates in the PDF
this.numFields; // number of fields in the file
```

45 Autotabbing Fields

You can set up fields with autotabs so that when a user completes filling in a field, the cursor jumps to the next field. You might use this feature if you have a SS number or phone number where the numbers are filled in separate fields.

To set up an auto tab, do the following:

- ▶ Write a Document Level JavaScript and enter the following code:
 1. `function goNext(item, event, cName)`
 2. `{`
 3. `AFNumber_Keystroke(0, 0, 0, 0, "",true);`
 4. `if (event.rc &&`
 5. `AFMergeChange(event).length ==event.target.charLimit)`
 6. `item.getField(cName).setFocus();`
 7. `}`
- ▶ Create three fields (*ss.1*, *ss.2*, *ss.3*) and set the limit of characters in the Options tab to 3 for the first field, 2 for the second and 4 for the third. We'll use a Social Security number as an example.
- ▶ Open the Format tab in the first field properties and click the Custom Keystroke Script radio button and click Edit.
- ▶ In the JavaScript Editor, add the following code.
 1. `goNext(this, event, "ss.2");`

In the above script, *ss.2* is the next field that follows the field where the script is written.
- ▶ Repeat the same steps for the second field and jump to the next field (*ss.3*).

Printing Pages Via JavaScript

Adding buttons to print pages is particularly useful when your forms are displayed in Full Screen mode.

To write a script that prints the current page in view, use the following script on a button field:

```
1. this.print (false, this.pageNum, this.page-
   Num, {bShrinkToFit: true});
```

The *false* item in the above script prevents the Print dialog box from opening and the page is sent directly to the printer. The *{bShrinkToFit: true}* item shrinks the page to fit within the paper size if the PDF page is larger than the output page size.

To print all pages in a document, use:

```
1. this.print(true);
```

Printing from the JavaScript Console

If you happen to be viewing a file in Full Screen mode and you want to print a page or the entire document while remaining in Full Screen mode, you can use the JavaScript Console.

To print a document while in Full Screen mode, do the following:

- ▶ Press Command/CTRL + J to open the JavaScript Editor.
- ▶ In the JavaScript Console type the following code to print the entire document.

```
1. this.print(true);
```

- ▶ Press the NumPad Enter key on your keyboard.



48

Eliminating Fields From Print

You may want to print a document without the data in certain fields printed on the output. You can use the Form Field drop down menu in the General tab in the Document Properties and choose *Visible but doesn't print* and the field won't print on the output.

If you have some conditional items where if *X* occurs, then you don't want certain fields to print, you can use a JavaScript to control whether fields print or not. To prevent a field from printing, use the following example:

- ▶ On a button field create a JavaScript and enter the following code:

```
1. var f = this.getField(myField");
   // myField is the field name
   f.print = false;
```

49

Printing with Annotations

If you want to print a file with the annotations added to the output, use the following JavaScript on a button field:

- ▶ Create a button field and add a JavaScript using the following code:

```
1. this.print (true, {bAnnotations: true,
   bShrinkToFit: true, bReverse: true});
```

- ▶ The above code is used to print annotations and the *bShrinkToFit* toggle is on as well as printing in reverse order. To eliminate any of the items, delete those not desired from the code.

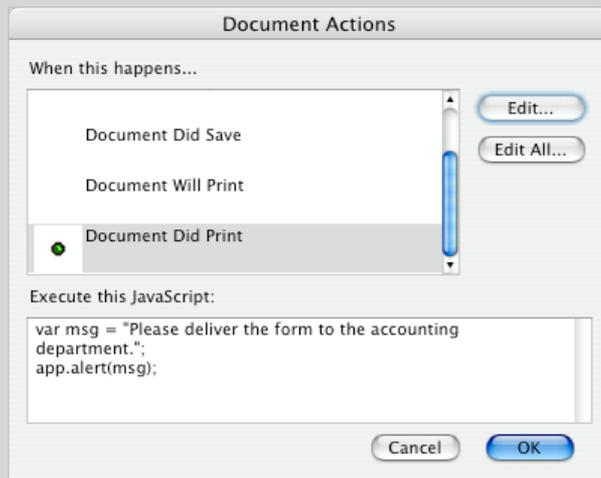
Creating Document Actions

You can execute JavaScripts during certain document actions such as saving a file, printing a file, or closing a file. You may want to add an application alert when a file prints to provide the user instructions on what to do with the form after printing it.

To write a script on a Document Action, do the following:

- ▶ Choose Advanced | Document Processing | Set Document Actions.
- ▶ Choose an action in the scrollable list (for example Document Did Print).
- ▶ Click Edit to open the JavaScript Editor.
- ▶ Add an application alert in the JavaScript Editor such as the example shown:

1. `var msg = "Please deliver the form to the accounting department";`
2. `app.alert(msg);`



51

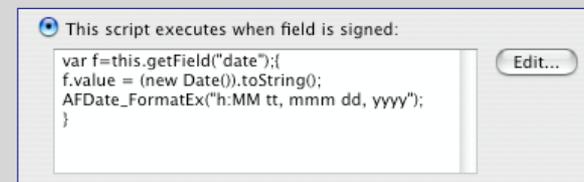
Date Stamping a Document

You may want to date stamp a form at the time a user signs the form or submits the form.

You can add a JavaScript on a signature field or a button. Use the following example to add a date stamp when a user signs a form:

- ▶ Create a Signature field on a form and click the Signed tab in the Signature Properties dialog box.
- ▶ Click the *This field executes when the field is signed* radio button and click the Edit button.
- ▶ Type the following script in the JavaScript Editor:

```
1. var f=this.getField("date");{
    f.value = (new Date()).toString();
    AFDate_FormatEx("h:MM tt, mmm dd, yyyy");
}
```



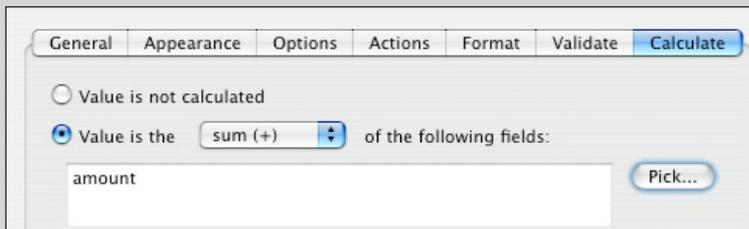
date in the script above is the name of the field where the date will be added.

Summing Rows and Columns

The easiest way to sum a column or row is to *not* write a JavaScript and use the built-in formula for summing data that Acrobat provides you. To use the built-in formula for summing a series of data fields, do the following:

- ▷ Create a column (or row) of fields formatted as numbers.
- ▷ Be certain to name the fields with hierarchical names such as *amount.0*, *amount.1*, *amount.2*, etc.
- ▷ Double-click the field that holds the data sum with the Select Object tool to open the field properties dialog box.
- ▷ Click the Calculate tab.
- ▷ Choose *sum(+)* from the *Value is the* drop down menu.
- ▷ Click the *Pick* button.
- ▷ Check the box where you see the parent name in the Field Selection dialog box.
- ▷ Click OK.

When you return to the Calculate pane in the field Properties, you should see the parent name shown in the field box below the *Value is the* drop down menu.



53

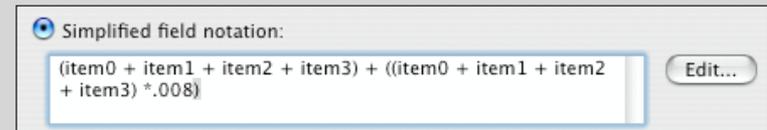
Using Simplified Field Notation

Acrobat offers only a limited number of built-in formulas. If you want math functions such as division and subtraction, you need to use either JavaScript or Simplified Field Notation.

Simplified Field Notation is a much easier scripting language based on syntax similar to formulas written in spreadsheet programs. Keep in mind that you *cannot* use hierarchical field names when using Simplified Field Notation.

To calculate a sum and add a sales tax using SFN, do the following:

- ▷ Open a text field properties for a total field and click Calculate.
- ▷ Click the *Simplified field notation* radio button and click the Edit button.
- ▷ Type the following script in the JavaScript Editor:
 1. $(item0 + item1 + item2 + item3) + ((item0 + item1 + item2 + item3) * .00725)$
- ▷ The above script sums data in fields *item0*, *item1*, and *item2* and adds a sales tax of 7.25%. Note that the fields all have separate names and do not use parent/child names.



Summing Data with JavaScript

You may want to sum data in a column or row and add a sales tax to the sum, a shipping charge, apply a discount, or some other math function as part of the field calculation.

If you use the built-in formula described in Tip #52 you can't add a second calculation. If you use a Simplified Field Notation script, you need to name each field separately in the formula —something that could be very tedious with long columns or rows.

In order to extend the formula to calculate a total and apply the total to another calculation, your best option is to use JavaScript.

- ▶ Open the Calculate tab on a field to hold the data calculation and click Edit for *Custom Calculation Script*.

- ▶ In the JavaScript Editor, type the following code:

```
1. var f = this.getField("amount");
2. var a = f.getArray();
3. var sum = 0;
4.  for (i =0; i < a.length; i++)
5.    sum += a[i].value;
6. event.value = sum + (sum * .08);
```

Line 1 sets up an array with a field having a parent name of *amount*. The form contains several fields named *amount.0*, *amount.1*, *amount.2*, etc. The loop is executed and picks up the value for each field and adds it to a variable called *sum*.

The last line adds the *sum* to the *sum* * a sales tax of 8%.

55

Calculating a Sales Tax

Sales tax calculations can be handled easily with built-in formulas, Simplified Field Notation, and JavaScript when you have separate fields for a subtotal and a tax field.

Using JavaScript the following script would be used to calculate a sales tax:

- ▶ Open the Calculate tab on a field to hold the data calculation and click Edit for *Custom Calculation Script*.

- ▶ Type the following script in the JavaScript Editor:

```
1. var f = this.getField("subtotal");
2. event.value = Math.round(f.value * 7.25) / 100;
```

- ▶ The above script assigns a variable to a field called *subTotal* where the sum of data in a row or column has been calculated. The second line of code calculates a sales tax of 7.25%.

56

Calculating a Shipping Charge

A shipping charge might be calculated based on the total cost of items purchased. The following formula calculates a shipping charge of 12% of the total purchase amount.

- ▶ Open the Calculate tab on a field to hold the data calculation and click Edit for *Custom Calculation Script*.

- ▶ Type the following script in the JavaScript Editor:

```
1. var g = this.getField("subtotal");
2. event.value = Math.max(2, Math.floor(g.value / 12));
```

- ▶ The above script assigns a variable to a field called *subTotal* where the sum of data in a row or column is calculated. The second line of code calculates a shipping charge of 12%.

Importing Images

You may want to add a photo or logo to a form and enable the user to import an image on a button action.

Here's how to do it:

- ▶ Create a button field and choose Icon only in the Options tab.
- ▶ Click the Actions tab and add a JavaScript.
- ▶ In the JavaScript Editor, type the following code:

```
1. event.target.buttonImportIcon();
```

When the user clicks the button, the Select Icon dialog opens.

Note: to make this work in Adobe Reader see Tip # 88.

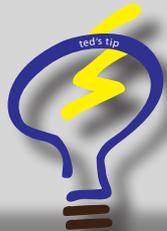


Showing/Hiding Fields

Acrobat provides you an action to show and hide fields. If you have many fields to show/hide, you need to check each item in the Show/Hide Field dialog box. An easier way to handle the task is to use JavaScript. Be certain to use hierarchical names before writing the script.

- ▶ Create a button field and click the Actions tab and add a JavaScript.
- ▶ In the JavaScript Editor, type the following code:


```
1. var f = this.getField("myField");//parent name
2. f.hidden=true;//hides all fields with the same parent name
```



To show hidden fields use the following script in line 2:

```
2. f.hidden=false;
```

59

Resetting a Form

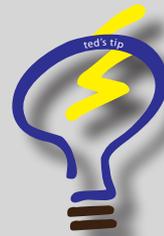
Acrobat has a built-in function for resetting a form that was described in Tip #21. This action works fine when you want to reset all fields or all fields with a few exceptions on a form.

You may have different sections of a form that you don't want to reset such as the identifying information or fields that contain fixed data that you use for calculations.

Using JavaScript the following script would be used to reset form fields having the same parent name:

- ▶ Create a button and add a JavaScript.
- ▶ Type the following script in the JavaScript Editor:


```
1. var resetFields = this.getField("customerData");
2. this.resetForm(resetFields);
```
- ▶ The above script resets the fields with the parent name of *customerData*.



If you want to reset all fields on a form, use the following script:

```
this.resetForm();
```

This kind of script might be added as a Document Level JavaScript or Page Action script along with an Application Response dialog box that prompts the user to reset a form.

If a user partially completes an order form, the user would say no to resetting a form. If the user starts anew, then perhaps all the data might be reset on the form.

Setting Fields to Read Only

Fields that are set to Read Only in the General tab of the field properties dialog box, prevent data entry or changing the field data by an end user. If the form is secured with Acrobat Security, the user won't be able to edit the fields and change the data.

You might set fields to Read Only that have fixed price values and calculations.

To set a field for Read Only, do the following:

- Open the properties dialog box on any field.
- Click the General tab.
- Check the box for Read Only.



You might start with fields that are marked for reading and writing and later, after the user executes certain actions, you might want to lock the fields to prevent data entry. To handle this task add a JavaScript in the Calculate tab where you want the field marked as Read Only.

Here's a sample:

```
1. var f = this.getField("item")
2. f.readonly=true;
```

All fields with the same parent name are marked as Read Only.

To disable Read Only, use the following in a script like the above:

```
2. f.readonly=false;
```

61

Deleting Fields

Deleting fields can provide another level of security on a form. When a user submits a form or prints a form, you can delete all fields with no data. Something like a travel reimbursement form might be one example of a form where deleting fields could be employed.

If there are no fields to add data, the form can't be changed by another user.

Using JavaScript the following script would be used to delete a series of fields containing no data:

- Add a JavaScript to a Signature Field, a Document Action, or a Submit button.
- Type the following script in the JavaScript Editor:
 1. `for (var i = 0; i < 5;i++)`
 2. `{var f = this.getField("amount."+i);`
 3. `if (f.value == "")`
 4. `removeField("amount."+i);`
 5. `}`
- The above script deletes all fields with the parent name of *amount* that have no data.



Deleting Zeros from Calculation Fields

If you have a table and the last column or row of data is a calculated field (something like price * qty), the default result is *0.00* until data are added for the fields within the calculation.

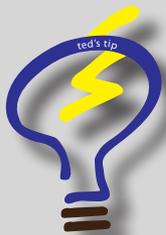
You can eliminate default zeros in calculated fields with JavaScript.

Example script:

```
1. var f = this.getField("price.0");
2. var g = this.getField("qty.0");
3. if (g.value !=0)
4.   event.value = f.value * g.value;
5. else
6.   event.value = "";
```

In lines 1 and 2 the variables are assigned for a price and quantity field. The result of the calculation is an amount determined by multiplying the price * the quantity.

Line 3 checks the *qty* field and if it's not zero, the calculation is performed. If the result is zero, line 6 is executed where no value is entered in the field resulting in eliminating default zeros.



Note that you cannot eliminate zeros using the built-in formulas or Simplified Field Notation. You must use JavaScript in order to eliminate the zeros.

63

Adding URLs to Text

Text added to a background design that contains a URL or email address is automatically recognized in Acrobat viewers 7 and above. There's no need to add links or buttons to navigate to a URL or open your default email client on a button click.

However, if you ask a user to add a URL to a text field, the text field isn't recognized as a URL link. Using JavaScript, you can create the link by doing the following:

- Create a Text field.
- Open the Actions tab and choose Mouse Up for the Mouse Trigger. This trigger will execute the action when a user clicks on the text field.
- Add a JavaScript in the Actions tab and type the following in the JavaScript Editor:
 1. `if (!/^\s*$/ .test(event.target.value))`
`app.launchURL(event.target.value);`
- The next time a user clicks in the text field, the URL opens in the user's default Web browser.

64

URL Links in New Windows

By default, a URL link opens in the same window if the browser is open and displays a web page. Using JavaScript you can force a URL link to open in a second window:

- On a link button or any field containing a URL link use the following script:

```
1. app.launchURL("http://www.mycompany.com/file.pdf", true);
```

The *true* statement in the above code triggers opening the link in a new browser window.

Adding Annotations Using JavaScript

You might have a need for a user to add a comment note, add a Stamp comment, or a Text Box comment based on a conditional response.

Perhaps clicking a check box where the item is *Other* among a list of other options is one example where you might use this feature. If the check box is clicked, a Text Box is added to the document page or some such other type of condition.

Note that the file must be enabled with Adobe Reader usage rights for this to work in Adobe Reader.

- ▶ Add a JavaScript to a button, a check box, or radio button and click the Actions tab.
- ▶ Type the following in the JavaScript Editor:

```
1. var annot = this.addAnnot(
2.   {page: this.pageNum,
3.   type: "FreeText",
4.   rect: [396,298,629,406],
5.   strokeColor: color.white,
6.   fillColor: ["RGB", .85, .85, .85],
7.   }
8. );
```

Line 3 identifies the type of annotation to be added to the page. In this example *FreeText* is a Text Box comment.

Line 4 specifies the Text Box rectangle coordinates. You need to precisely determine where the coordinates are for the annotation you want to add using the script (See Tip #68 for how to determine x,y coordinates).

Lines 5 and 6 set attributes for the border color and the fill color of the Text Box.

66

Adding Fields Using JavaScript

You can add fields using JavaScript. You might add a field for user input based on a user response.

This task cannot be performed in Adobe Reader when forms are created in Acrobat. However, you can create dynamic forms in Adobe LiveCycle Designer that enable Adobe Reader users to spawn fields on a page.

Example script:

```
1. var f = this.addField("newField", "text",
2.   page = this.pageNum, [100,100,200,200])
3.   f.textSize = 9;
4.   f.textfont = font.Helvetica;
5.   f.textColor = color.white;
6.   f.fillColor = new Array("RGB", .9, .9, .9);
7.   f.alignment = "left";
8.   f.multiline = true;
9.   f.readonly = false;
```

Line 2 identifies the x,y coordinates for the field rectangle. Before writing the script, be certain to identify the exact coordinates where you want the field to appear.

See Tip #67 for an easy method for identifying x,y coordinates.



Determining x,y Coordinates

Tips #65 and #66 use x,y coordinates to determine the bounding area for annotations and fields. Before you start writing a script you need to know the coordinates for where you want the object to appear on the page.

To easily determine the coordinates you'll use in your JavaScript code, do the following:

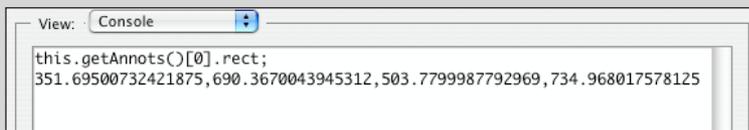
- Open the Comment and Markup toolbar.
- Click the Rectangle Comment tool.
- Draw a Rectangle comment in the area of your form where you want to create an annotation or field.
- Open the JavaScript Console (Command/CTRL + J).
- Enter the following code in the Console.

```
1. this.getAnnots()[0].rect;
```

- Press the NumPad Enter key on your keyboard.

The JavaScript Console reports back to you the coordinates.

- Ignore all digits after the decimal point and use the whole numbers for the x,y coordinates in your script.



```
View: Console
this.getAnnots()[0].rect;
351.69500732421875,690.3670043945312,503.7799987792969,734.968017578125
```



68

Changing Text Colors

By default, Acrobat permits you to identify negative values with a red color in the Format tab. If you want to emphasize a value that's not a negative value or use another color other than red, you can do so using JavaScript.

Example script in the Format tab in a text field:

```
1. var f = this.getField("fieldName");
2.   if (event.value >50)
3     f.textColor = color.red;
4. else
5.   f.textColor = color.black
```

The above script sets the text color to red if the value in a field named *fieldName* is greater than 50.



Acrobat provides you with some default colors that you can assign to text and objects. If you want to create a custom color look up a color in a program like Adobe Photoshop and record the RGB (or CMYK) values. The colors in Photoshop use whole numbers to express color.

In Acrobat you define a color by using something like ["RGB", .503, .982, .495]

To convert the color values from Photoshop to Acrobat, take a Photoshop color and divide by 255. Therefore a color like a red value of 149 in Photoshop would be a red value of .584 in Acrobat. The syntax for changing a color in a field would be something like:

```
event.target.textColor = ["RGB",
    .503, .982, .495];
```



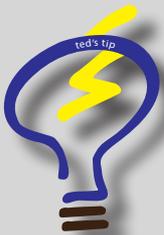
Spawning Pages From Templates

Spawning pages from templates provides you with options for many uses. PDF files can be made more dynamic by adding content on-the-fly through new pages added to a document.

- ▶ Add a page to a document that you want to use for a template and navigate to that page.
- ▶ Click the Edit Layout tool in the Forms toolbar.
- ▶ Choose Forms | Page Templates.
- ▶ Type a name in the Page Templates dialog and click *Add*. To hide the template from view, click the Eye icon.
- ▶ Add a JavaScript to a button field and enter the following code in the JavaScript Editor.

```
1. var a = this.getTemplate("myTemplate");
2.   a.spawn ({
3.     nPage:this.numPages,
4.     bRename:true,
5.     bOverlay:false
6. })
```

Be certain that you use the name you assigned to the template in Line 1. When you click the button, a new page is spawned from the template.



By default the `bOverlay` switch is off meaning new pages are appended to the document. If you want to overlay the new page on an existing page (like adding a watermark), set the toggle to `true` as shown below:

```
    bRename:true,
```

70 Changing OCG States

Optional Content Groups are layers in a PDF document. You have available an action to Set Layer Visibility, but there are conditions where the action won't work and you need to use a JavaScript to change layer visibility.

To examine scripts that change layer visibility open the Attachments panel and open the file attachment: *housePlansForm.pdf*. Look over the JavaScripts in this document to discover how to change layer visibility using JavaScript.

71 Limiting Character Strings

You can limit the number of characters added to a text field with a JavaScript. To do so, use the example below:

- ▶ Open the Text field properties on a field.
 - ▶ Click the Format tab.
 - ▶ Click the Custom Keystroke Script radio button.
 - ▶ Click Edit and add the following script in the JavaScript Editor.
- ```
1. var f = this.getField("myField");
2. if (event.value.length > 10)
3. {
4. var msg = "Your entry is too long
 (limit 10 chars).";
5. app.alert(msg);
6. }
```

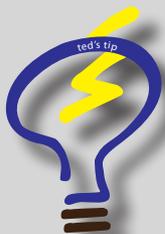


## Popup Menus for URL Navigation

Popup menus provide you with navigation options that can be used when viewing PDF files in Full Screen views. The following popup menu item opens different URLs in your default Web browser.

- ▶ Add the following script to a button:
  1. `// Declare pop-up menu properties as an array.`
  2. `var aParams = [`
  3. `{cName: "Acrobat User Community", cReturn: "www.acrobatusers.com"},`
  4. `{cName: "-"},`
  5. `{cName: "Acrobat User Blogs", cReturn: "http://www.acrobatusers.com/blogs"},`
  7. `{cName: "Ted's Blog Site", cReturn: "http://www.acrobatusers.com/blogs/tedpadova"}`
  9. `];`
  10. `var cChoice = app.popUpMenuEx.apply( app, aParams );`
  11. `if ( cChoice != null ) app.launchURL(cChoice);`

The above popup menu works much better than bookmarks if you intend to display your form in Full Screen mode.



You can add an unobtrusive popup menu to a form that provides URLs where users can explore more product information, contact info, company brochures, etc. One button can open many links and not disturb the form design.

# 73

## Popup Menus for Page Views

The `app.popUpMenuEx` object is the preferred way to write a script for application popup menus. However, you can get by with a much easier script that works well in all Acrobat viewers.

The following example uses a simple method for navigating pages in PDF.

- ▶ Navigate to a view you want for a page link to a popup menu item.
- ▶ Choose View | Navigation Panels | Destinations.
- ▶ Click the Create new destination tool in the Destinations panel.
- ▶ Repeat the steps for creating destinations to all views you want to link to in a popup menu.
- ▶ Create a button field and add a JavaScript with the following code:
  1. `var cChoice = app.popUpMenu`
  2. `(["First Page", "Home"],`
  3. `["Second Page", "Samples"],`
  4. `["Third Page", "JavaScripts"],`
  5. `["Fourth Page", "URL Links"],`
  6. `["Fifth Page", "Sample Forms"]);`
  7. `this.gotoNamedDest(cChoice);`

The first item in steps 2-6 is a parent name that shows a child name when the menu is opened. The child names are the same exact names as the destination names and a click on the name in the menu opens the destination view.

## Popup Menus for Opening Files

To create a popup menu that opens files, be certain to place the files in the same folder as the document containing the popup menu.

Add the following JavaScript to a button:

1. `var cChoice = app.popUpMenu`
2. `(["Human Resource Forms", "hrForm.pdf", "vacationLeave.pdf"],`
3. `["Sales Forms", "salesOrder.pdf", "customerID.pdf", "purchaseForm.pdf"],`
4. `["Administration Forms", "creditApplicationForm.pdf", "creditApprovalLetter.pdf"]);`
5. `this.slave = app.openDoc((cChoice), this);`

Each of the items (2-4) has multiple submenu items that open the respective PDF documents.

Don't forget the semicolon (;) at the end of line 4. Each of the other lines of code (2-3) end with a comma.

## Emailing a PDF

From a button or a JavaScript assigned to another action you can attach the open PDF document to an email message.

On a button field, add the following script:

1. `// use mailDoc in this routine`
2. `this.mailDoc(true, "yourname@address.com", "cc goes here", "bcc goes here", "Subject goes here");`

# 76

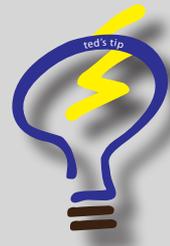
## Emailing Form Data

From a button or a JavaScript assigned to another action you can attach just the form data from the open document to an email message.

On a button field, add the following script:

1. `// use mailForm in this routine`
2. `// gets the user address from a field`
3. `var f = this.getField("useraddress");`
4. `this.mailForm(true, f.value, "", "", "Subject");`

The above script attaches an FDF form data file to an email message. Line 3 picks up an email address from the form that the PDF author added to a text field and uses that address for the *To* field in a new email message.



Note that if you want to reference fields on a page that have been filled in by you and you don't want them to be visible to the end user, you can fill in a field and hide the field.

Open the General tab in the field properties dialog box and choose Hidden from the Form Field drop down menu.

JavaScript will pick up the field data even when the field is hidden.

## Checking for Empty Fields

Before a document is submitted, you might want to have the form checked to be certain that all fields have been filled in. The script to perform such a function has two parts. The first part is a function added as a document level JavaScript. The second part is the script that executes when the Submit a Form button is clicked.

Add the following function as a Document Level JavaScript and name the function *aField*:

- ▷ Choose Advanced | Document Processing | Document JavaScripts.
- ▷ Type *aField* in the Script Name text box and click Add.
- ▷ Type the following code in the JavaScript Editor.

```
1. function checkField(aField)
2. {
3. if (aField.value == "") // empty field
4. {
5. var msg = "No fields can be left empty.";
6. app.alert(msg);
7. return 0;
8. }
9. } // end of function
```

Click Close in the JavaScript Functions dialog box.

On a submit button, add the following script:

```
1. // the procedure begins here
2. var okToSubmit = true;
3. // loop over all fields:
4. for (var j = 0; j < this.numFields; j++)
5. {
6. var fieldname =
7. this.getNthFieldName(j);
8. var theField = this.getField(fieldname);
9. if (theField.type != 'text')
10. continue; // get past the button fields
11. var valid = checkField(theField);
12. if (!valid) // valid == 0? Halt!
13. {
14. okToSubmit = false; // set flag
15. break; // exit loop prematurely
```

The above routine checks all text fields to be certain some data exist in the fields. If a field is left empty, the app.alert dialog box described in the *aField* function reports the error.



## Creating Fixed Response Options

You may have a form where you want a user to make a selection from a Combo Box and based on that selection, offer choices respective to the first choice in a second Combo Box.

Something like choosing a shirt in one column and have sizes appearing in a second column might be one example of such a case. If you have shirts, pants, hats, and other articles of clothing, the choices in the second column will change based on the first column choice.

For the script (opposite page) to execute properly, you need to create a table with at least two columns (or rows). The script is added to the fields in the first column (or row) in the Format tab, then duplicated for all subsequent columns/rows.

Do the following to set up the table:

- ▶ Create a table with 5 columns as shown below.
- ▶ Open the text field properties in the first field in column one and click the Format tab.
- ▶ Add a Custom Keystroke Script and type the code at right in the JavaScript Editor.

| Item  | Type | Price | Qty | Amount |
|-------|------|-------|-----|--------|
|       |      |       |     |        |
|       |      |       |     |        |
|       |      |       |     |        |
| Total |      |       |     |        |

```

1. if (!event.willCommit) {
2. var a = this.getField("type.0");
3. a.clearItems();
4. switch (event.change) {
5. case "Golf Balls":a.setItems(["",
 "Titleist", "Pinnacle", "Maxfli"]);
6. break;
7. case "Golf Hats":a.setItems(["", "Navy",
 "White", "Green"]);
8. break;
9. case "Golf Shirts":a.setItems(["", "Small",
 "Medium", "Large", "Extra Large"]);
10. break;
11. case "Golf Shoes":a.setItems(["", "Size 9",
 "Size 9.5", "Size 10", "Size 10.5",
 "Size 11", "Size 11.5", "Size 12"]);
12. break;
13. case "Golf Tees":a.setItems(["", "25",
 "50", "100"]);
14. break;
15. }

```



If you want a fixed price for each item selected for *Item* and *Type*, add a JavaScript in the Calculate tab for the *Price* field.

Open a price field in the table at left and you can view the JavaScript added here to create the fixed prices.

## Creating Application Response Dialog Boxes

An application response dialog box typically presents the user with a question and solicits an answer. Based on the response certain actions can be invoked. You might want to ask a user how many products to purchase. The user responds and you can then take the data from the dialog box and add it to a field object.

To perform such a task, do the following:

- ▶ Create a text field.
- ▶ Create a button field and add the JavaScript below:
  1. `var t = this.getField("qty");`  
`// the target field`
  2. `var cResponse = app.response({`
  3. `cQuestion: "How many do you want?",`
  4. `cTitle: "Quantity to purchase"});`
  5. `if ( cResponse == null){`  
`// if Cancel is selected`
  6. `app.alert("Cancel order for this item?");`
  7. `cResponse = 0;`
  8. `}`
  9. `else`
  10. `app.alert("You ordered, \""+cResponse+"\"... items",2);``//confirmation dialog`
  11. `{`
  12. `t.value = cResponse;` `// places the data from the dialog to the target field`
  13. `}`

Click Me

# 80

## Adding Data to Secondary Forms

If you have a product brochure of a large file size, you might want to add application response dialog boxes to populate a secondary form or copy data from a form page on a product brochure to another document you use as an order form.

On a button field, add the following script:

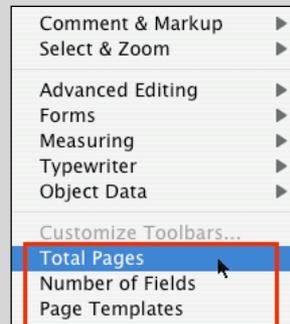
- ▶ Create a form and add the following Document Level JavaScript:
  1. `this.disclosed = true;`
- ▶ On a button field in another document add the following script:
  1. `this.slave = app.openDoc("myForm.pdf",this);`
  2. `this.bringToFront();`
  3. `for (var i = 0; i <4; i++) {``//good for 4 fields`
  4. `var quantity = this.getField("qty."+i);`
  5. `var result = this.slave.getField("oQty."+i);`
  6. `result.value = quantity.value;`
  7. `var amount = this.getField("total." +i);`
  8. `var cResult = this.slave.getField("oAmount."+i);`
  9. `cResult.value = amount.value;`
  10. `}`
  11. `slave.bringToFront();`



If you want Adobe Reader users to fill in a form apart from a brochure, add the form to a PDF Package. Add a submit form button and enable the form with Reader usage rights. When the Reader user submits a form in a PDF Package, only the form is sent back to you.



# Adding Menu Items



Scripts cannot be written in Adobe Reader, but the menus created in Acrobat do appear in Adobe Reader.

## About Adding Menu Commands



You can add custom menu commands in Acrobat and the same commands you create in Acrobat are also usable in Adobe Reader.

To create a custom menu command you need to add a JavaScript to your JavaScripts folder.

On Windows the directory path for the JavaScripts folder is:  
Program Files:Adobe:Acrobat: 8.0: Acrobat:JavaScripts

On the Macintosh the path is:  
Users:User Log-on:Library:Acrobat User Data:80\_x86:JavaScripts

Copy the file attachments in this document to the JavaScripts folder on your computer.

Save your text files with a .js extension.

## Counting Fields

Add a menu command that counts the total fields on a form. This can help compare forms, diagnose forms, and provide information on forms that have been secured with Acrobat Security.

When you select the menu item, a dialog box opens reporting the find.

- ▶ Write a JavaScript with a text editor using the following code:

```
1. function totalFields(){
2. app.alert("Total Fields: " + this.numFields);
3. }
4. app.addItem ({cName: "Number of Fields",
 cParent: "Tools", cExec: "totalFields()"}
5. });
```

The above script adds the menu command to the Tools menu (line 4).

# 82

## Counting Page Templates

Add a menu command in the Tools menu that counts Page Templates. Counting Page Templates can be helpful in diagnosing forms and provide information about forms that have been secured with Acrobat Security.

- ▶ Write a JavaScript with a text editor using the following code:

```
1. function totalTemplates(){
2. app.alert("Total Page Templates: " + this.
 numTemplates);
3. }
4. app.addItem ({cName: "Page Templates",
 cParent: "Tools", cExec: "totalTemplates()"}
5. });
```

The above script adds the menu command to the Tools menu (line 4).



You can add both menu commands in Tips #81 and #82 in the same script. Copy lines 1-3 in one script and paste it after line 3 in the first script. Then paste lines 4-5 at the end of the first script.

Open the Attachments pane in this file and copy the tools.js file to your JavaScript folder and you won't need to write the scripts.

## Listing Menu Items

If you want to access a menu command using JavaScript or add a menu item, you need to know the menu items names used by JavaScript.

To view a list of menu item names, do the following:

- ▶ Open the JavaScript Console (Command/CTRL + J).
- ▶ Type the following code in the Console:
  1. `app.listMenuItems();`
- ▶ Press the NumPad Enter key and all the menu item names are shown in the JavaScript Console.

```
app.listMenuItems();
[[cName: Apple, oChildren: [[cName: About], [cName: AboutAdobeExtensions], [cName:],
[cName: GeneralPrefs], [cName:], [cName: Services], [cName: HideAcrobat],
[cName: HideOthers], [cName: ShowAllApps], [cName:], [cName: Quit]], [cName: File,
oChildren: [[cName: New, oChildren: [[cName: Letter Portrait], [cName: Letter Landscape],
[cName: A4], [cName: Slide (10x7.5)], [cName: Screen (640x480)], [cName: Note (4x3)],
[cName: Custom...]]], [cName: Open], [cName: Organizer, oChildren: [[cName: OpenOrganizer],
[cName: AddToOrganizer], [cName: OrganizerNewCollection], [cName: OrganizerFavorites],
[cName:], oChildren: [[cName:]], [cName:], oChildren: [[cName:], [cName:], [cName:]],
[cName:]], [cName: OpenRecent, oChildren: [[cName: OrganizerHistory, oChildren:
[[cName:], [cName:], [cName:], [cName:], [cName:], [cName:], [cName: Separator],
[cName: OpenOrganizer]], [cName: RecentFile1], [cName: RecentFile2], [cName: RecentFile3],
[cName: RecentFile4], [cName: RecentFile5], [cName: RecentFile6], [cName: RecentFile7],
[cName: RecentFile8], [cName: RecentFile9]], [cName: endOpenGroup], [cName: NewDocument,
oChildren: [[cName: NewDocFromFile], [cName: NewDocCombineFiles], [cName: Scan],
[cName: Web2PDF: OpnURL], [cName:], [cName: ScreenCapture_Screen],
[cName: ScreenCapture_Window], [cName: ScreenCapture_Selection],
[cName: ImageConversion: Clipboard], [cName: endNewDocumentGroup], [cName: Editor: New]],
[cName: CombineFiles], [cName: endNewDocumentOnlineGroup], [cName: Breeze: StartMeeting],
[cName: Breeze: StartMeetingSeparator], [cName: Save], [cName: SaveAs],
[cName: DIGSIG: SaveAndAuthenticate], [cName: SaveACopy], [cName: File.Export, oChildren:
[[cName: Export.com.adobe.acrobat.doc], [cName: Export.com.adobe.acrobat.rtf],
[cName: Export.com.adobe.acrobat.xml-1-00], [cName:], [cName: Export.avExportMenu_HTML,
```



To display toolbar buttons, type the following in the JS Console:

```
app.listToolbarButtons();
```

# 84 Adding a URL to a Menu

You can add menu commands in Acrobat that open URLs in your default Web browser.

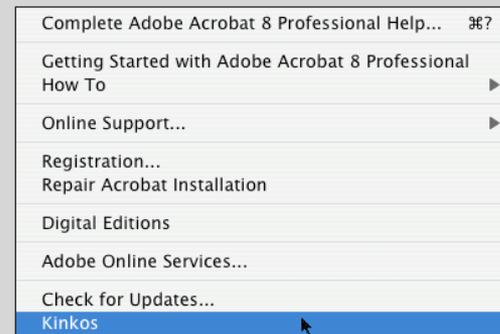
Here's the JavaScript code for accessing the FedEx/Kinkos web site to order prints. Replace the URL in this script with your favorite print service.

- ▶ Write a JavaScript with a text editor using the following code:

```
1. app.addItem({
2. cName: "FedExKinkos",
3. cParent: "Help",
4. cExec: "app.launchURL('https://partner.fedexkinkos.com/partnerInit.do');",
5. nPos: 16
});
```

Line 5 in the above code positions the command in the Help menu. You may want to change the position by typing a different value.

When the menu command is selected your default Web browser opens and takes you to the target URL.



## Add a File | New Menu Command

Acrobat provides you an option to create new blank pages. However, the new pages by default have a text block that can be annoying when you want to use blank pages for temporary documents to hold form fields.

With a little JavaScript you can create the missing File | New command that Acrobat has avoided through 8 generations of the program. The following code creates the command with several different custom page size options:

► Type the following code in a text editor and save the file to your JavaScript folder with a .js extension:

```

1. trustedNewDoc = app.trustedFunction(function
 (nWidth, nHeight)
2. {
3. app.beginPriv();
4. switch(arguments.length) {
5. case 2:
6. app.newDoc(nWidth, nHeight);
7. break;
8. case 1:
9. app.newDoc(nWidth);
10. break;
11. default:
12. app.newDoc();
13. }
14. app.endPriv();
15. })

```

```

16. app.addSubMenu({ cName: "New", cParent:
 "File", nPos: 0 })
17. app.addMenuItem({ cName: "Letter Portrait",
 cParent: "New", cExec: "trustedNewDoc();" });
18. app.addMenuItem({ cName: "Letter
 Landscape", cParent: "New", cExec:
 "trustedNewDoc(792,612);" });
19. app.addMenuItem({ cName: "A4", cParent: "New",
 cExec: "trustedNewDoc(420,595);" });
20. app.addMenuItem({ cName: "Slide
 (10x7.5)", cParent: "New", cExec:
 "trustedNewDoc(720,540);" });
21. app.addMenuItem({ cName: "Screen
 (640x480)", cParent: "New", cExec:
 "trustedNewDoc(640,480);" });
22. app.addMenuItem({ cName: "Note
 (4x3)", cParent: "New", cExec:
 "trustedNewDoc(288,216);" });
23. app.addMenuItem({ cName: "Custom...", cParent:
 "New", cExec:
24. "var nWidth = app.response({ cQuestion:'Enter
 Width in Points',\
25. cTitle: 'Custom Page Size'});"
 +"if (nWidth == null) nWidth = 612;"
 +"var nHeight = app.response({
 cQuestion:'Enter Height in Points',\
26. cTitle: 'Custom Page Size'});"
 +"if (nHeight == null) nHeight =
 792;"+"trustedNewDoc(nWidth, nHeight) "});

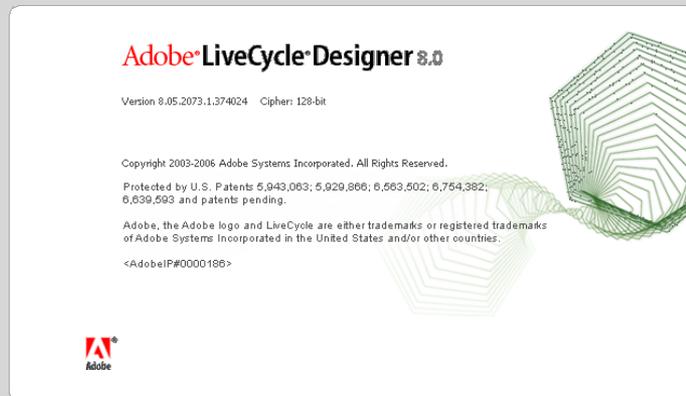
```



Open the Attachments panel in this file and copy the newPage.js file to your JavaScript folder and you won't need to write the script.

# Adobe LiveCycle Designer

Windows Only

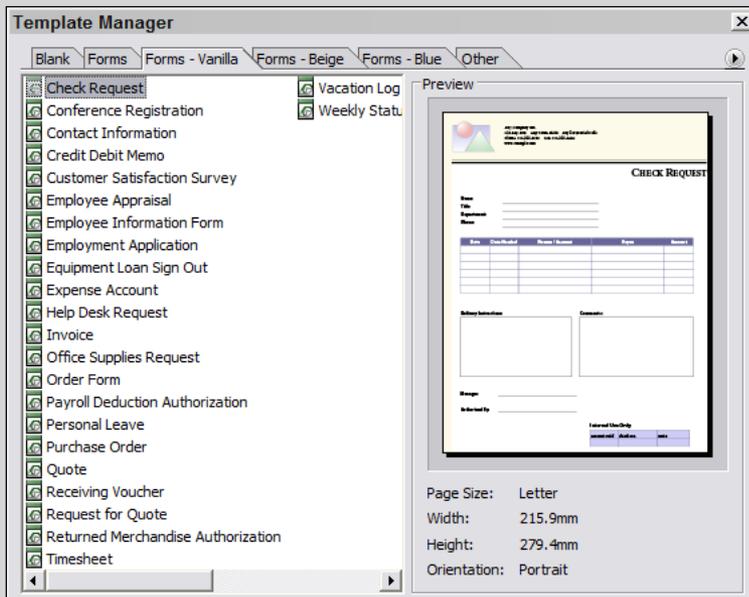


## Changing the Default Template

LiveCycle Designer (LCD) offers you an number of templates that you can modify to create your own personal forms. When you choose File | New the New Form Assistant opens where you can choose from one of the many templates available to Designer.

When you click the New tool in Designer, you can bypass the New Form Assistant dialog box and create a form based on a template. To change the default template in Designer to one that you prefer when clicking the New tool, do the following:

- ▶ Choose Tools | Template Manager (note that you do not need a file open to work in the Template Manager).
- ▶ Click the tabs to view the templates and choose the one you want to use as your default template.
- ▶ Click *Set Selected as Default* in the Template Manager.



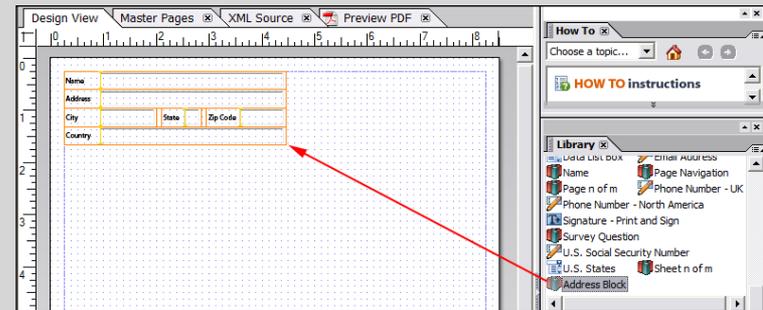
# 87 Adding an Address Block

Tip #27 described creating an address block in Acrobat by copying/pasting fields between documents.

LiveCycle Designer provides you a default block of fields you can drag and drop to your forms.

To add an address block to a Designer form, do the following:

- ▶ Create a blank new form (click the New tool in the Designer toolbar).
- ▶ Open the Library panel and scroll down to the Custom section in the panel.
- ▶ Click the Address Block item and drag it to your form.



ted's tip

You can customize an address block or any other group of fields and add them to your Library panel.

Edit the fields on a blank new page and drag them to the Library panel. The Add Library Object dialog opens where you can add a name, description, and select which group you want to add the fields.

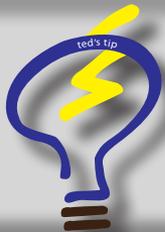
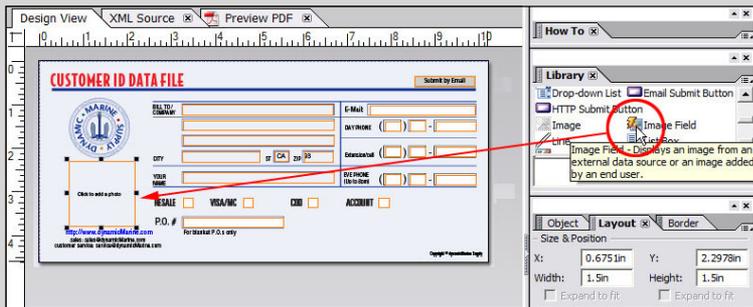
## Adding Button Imports

Tip #57 described how to add a button import on an Acrobat form using a JavaScript. When you create a button import in Acrobat, the JavaScript cannot be used in Adobe Reader.'

Using LiveCycle Designer, you can add a button field that an Adobe Reader user can use to add button faces to a button from a variety of different image formats.

To add a button that can be used to import images in both Acrobat and Adobe Reader, do the following:

- Create a form or import a form from Adobe Acrobat (see Tip #95 for instructions on converting Acrobat forms to LiveCycle Designer forms).
- Open the Library panel and open the Standard fields panel.
- Click the Image Field and drag it to your form.
- Save the form. Note that you do not need any scripting for the button action to work in Adobe Reader.



Note that the form does not need to be enabled with Adobe Reader usage rights for the Reader user to import images.

# 89

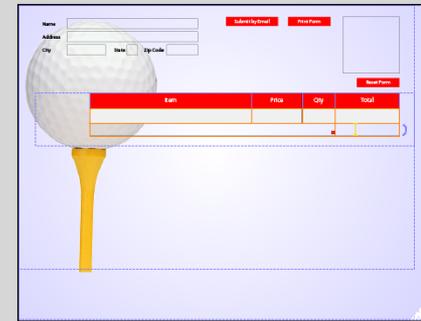
## Editing Backgrounds

LiveCycle Designer is not a sophisticated design tool as you find with programs like Adobe InDesign and Adobe Illustrator.

However, you can add a few design elements in Designer and change background colors to solid colors and gradients.

To change a background color to a gradient, do the following:

- Create a new blank document.
- Click the Master Page tab.
- Size down the default subform content area to reduce the size of the rectangle.
- Drag the Image object from the Standard library panel.
- Size the image to fill the page size.
- Open the Border panel.
- Choose from options in the Background fill for a Radial - to edge or Radial - to center.
- Resize the subform content area to the desired size.



## Binding an XML Sample File to a Form

Designing forms in LiveCycle Designer for data imports need to be debugged in two areas. You must have a proper XML file that can be imported in your form and you need to have the proper data connections and scripting on the form in Designer.

Testing and debugging a form can be a challenge if one or the other items is not properly set up. To test a form for data imports to be certain the form scripts work properly, you can start by creating a sample XML file.

To create a sample XML file you can use Windows NotePad. Another solution is to use an XML editor. My preference for editing XML files is using TextPad (<http://www.textpad.com>).

- Write a short script in an XML editor or Windows NotePad.
- In Designer, open the Data View panel on a new blank document.
- Open the Data View menu and choose New Data Connection.
- Choose Sample XML Data file in the New Data Connection dialog box and click Next.
- Select the sample XML file and click Finish.
- Drag fields from the Data View panel to the document page.

```

testTextPad.xml.txt
<?xml version="1.0" encoding="utf-8" ?>
<fields>
 <item>
 <field>Default Text</field>
 </item>
 <item>
 <field>Taylor Made Wood Set</field>
 </item>
 <price>
 <field>29.95</field>
 <field>999.95</field>
 </price>
 <qty>
 <field>2</field>
 <field>1</field>
 </qty>
 <total>
 <field>0.00</field>
 <field></field>
 </total>
</fields>

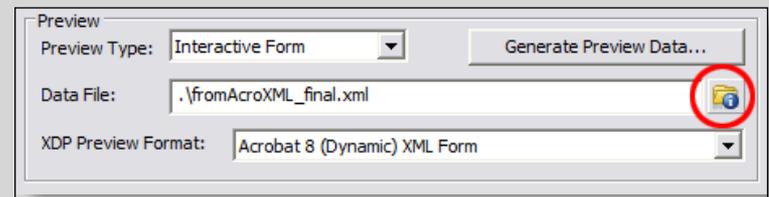
```

## 91 Testing the XML Data File

As a takeoff from Tip #90, continue editing your form with field formatting and layout.

When the form is designed as you like, open the Data View panel and do the following to test the data connection:

- Open the Data View panel and right click on Data Connection at the top of the menu to open a context menu.
- Choose Delete Data Connection.
- Remove the check mark in the Delete Data Connection dialog box and click OK.
- Choose File | Form Properties.
- Click the Defaults tab and click the Browse icon to the right of the Data File field box.



- Locate the sample XML file and click Open.
- Click OK in the Form Properties dialog box and click PDF Preview.

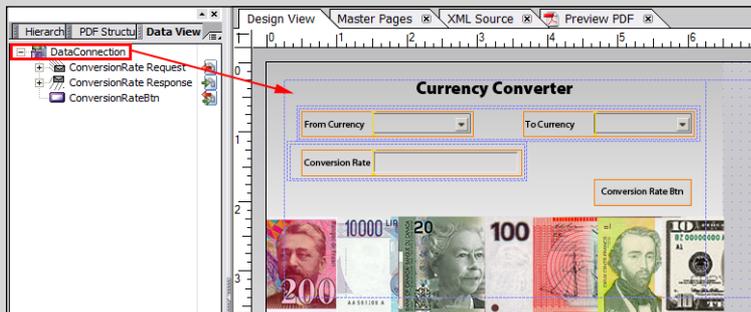
If the sample data do not import properly and if your XML file was written correctly, you'll know that you have some problems with the design of your form.

## Creating a WSDL Connection

A Web service can be searched on the Internet and you can bind data to a WSDL connection. You might want to connect to a weather service, a currency converter, or any number of different web services available on the Internet.

To connect to a Web service, do the following:

- ▶ Locate a URL for a Web service and one that you can bind to. You can search for Web services in your browser to poke around looking for a Web service.
- ▶ Copy the URL from your Location bar in your Web browser.
- ▶ In a new blank form in Designer open the Data View panel.
- ▶ Open the Data View menu and choose New Data Connection.
- ▶ Click WSDL File in the New Data Connection Dialog box and click Next.
- ▶ Paste the URL in the WSDL File text field and click Next.
- ▶ A list of operations appear in the New Data Connection dialog box. Look over the descriptions and choose the item to bind to by selecting it in Select an Operation list. Click Finish.
- ▶ Drag the Data Connection from the Data View panel to your form.



## 93

### Enabling an XML Form

Adobe Designer does not have a menu command for enabling PDF forms with Adobe Reader usage rights.

To enable a PDF with usage rights you need to use Adobe Acrobat.

- ▶ Complete your form editing in Adobe Designer.
- ▶ Save the file and open in Adobe Acrobat Pro (v8.1).
- ▶ Choose Advanced | Enable Usage Rights in Adobe Reader.

## 94

### Distributing an XML Form

Although a command exists in Adobe LiveCycle Designer for distributing a form, the wizard used with the Distribute form command is handled by Acrobat.

When you choose to distribute a form in Designer, Designer launches Acrobat and opens the form.

To distribute a form from Designer, do the following.

- ▶ Complete your form editing in Adobe Designer.
- ▶ Be certain you have a Submit button on your form.
- ▶ Choose File | Distribute Form.

The form opens in Acrobat and the Form Distribution Options wizard opens.

- ▶ Complete steps in the Form Distribution Options wizard to create a data set and email the form or save it and send later.

## Adding a PDF Background

In an effort to help Designer with a little more sophistication in the design stage of a form, version 8 supports adding PDF backgrounds.

If you want to add design appearances not available in LiveCycle Designer, create a design in a program like Adobe InDesign or Adobe Illustrator and export/save as PDF.

To import a PDF background, do the following:

- ▶ In Designer choose File | Open or click the Open tool.
- ▶ Locate the PDF document you want to import and select it in the Open dialog box. Click Open.
- ▶ Click the *Create an Interactive Form with Fixed Pages* radio button.

If you choose *Create an Interactive Form with a Flowable Content*, Designer can open the form, but the graphics will often change color values and may not appear like your original design. If using this option, you'll need to experiment a little on how to design a form so the content can be edited in Designer. Click Next.

Add an email button

Form fillers use the Email Submit button to return the form to the specified address.

Return email address:

Add a print button

Form fillers use the Print button to print the form and return it manually.

- ▶ Type an email address in the Return email address text box and click Finish.
- ▶ The design background is imported and you can now add the form fields in Adobe LiveCycle Designer.

# 96

## Converting a PDF Form

PDF forms with form data fields can be opened in LiveCycle Designer and the fields are recognized by Designer.

Before opening a form in Designer you need to decide whether it will be more efficient to use the original fields created in Acrobat or create new fields in Designer.

Assuming you want to use the fields created in Acrobat, do the following to convert the Acrobat form to a LiveCycle Designer XML form.

- ▶ In Designer, choose File | Open or click the Open button.
- ▶ Locate the PDF form containing data fields in the Open dialog box, select it, and click Open.

The Acrobat data fields are converted to XML form objects.

- ▶ Edit the form and save it.

The form is now an XML form.

Note that after saving the form, the form can no longer be edited in Adobe Acrobat.



If you have JavaScripts in your original PDF form, you'll need to rewrite all the scripts. Designer cannot translate the Acrobat JavaScripts to Designer JavaScripts.

Before converting a form, it's a good idea to delete all Document Level and Document Action JavaScripts.

## Using Transparent Images

Designer supports image formats that are limited to:

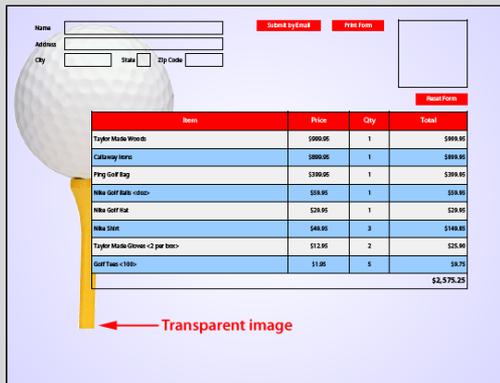
.bmp, .eps, .jpg, .gif, .png, and .tif.

Only those formats supporting transparency can be imported with the transparency intact.

As an easy method for importing transparent images in Designer, use Adobe Photoshop and choose the File | Save For Web & Devices command (Photoshop CS3). In the Save For Web & Devices window you can see what formats honor transparency.

Use either GIF or PNG-24 for the file format. EPS is supported by designer, but the rendering in Designer leaves much for desire. If your files are not intended for commercial press and color separating, GIF or PNG-24 will render the images fine onscreen.

- ▶ After saving a file in a format that preserves transparency, add the Image object to a form.
- ▶ Open a context menu on the Image object and select Choose Image.
- ▶ Select the image in the Browse for Image File dialog box and click Open.



# 98

## Calculating Columns

Tip #54 described a way to use an array to calculate a column or row of data using JavaScript.

Whereas Acrobat uses the scripting languages of Simplified Field Notation and JavaScript, Designer uses FormCalc and JavaScript.

Simplified Field Notation in Acrobat is not well suited for summing columns and rows of data, but FormCalc in Designer can be used quite easily for adding columns and rows of data.

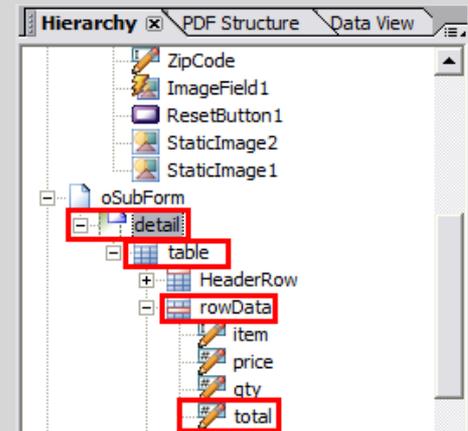
The following is an example of a FormCalc Script that sums data in a column:

1. `sum(detail.table.rowdata[*].total[*])`
- ▶ In the above script, *detail* is a subform containing a table.

*table* follows the subform and identifies the table where the data appear.

*rowdata* contains the cells in the table.

*total* is the field containing the data to be summed.



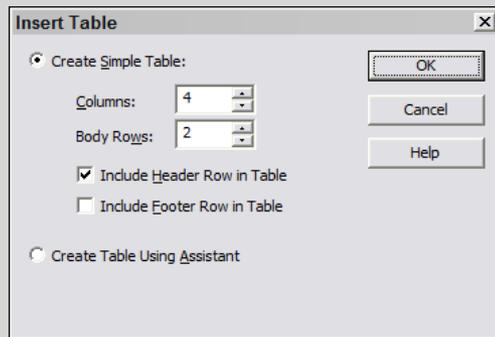
## Creating Tables

Tip #23 described creating a table of data using Acrobat. With Acrobat you need to design the table background in an authoring program if you want alternating rows of colors, apply stroke weights and colors, merge cells, and other features associated with table designs.

One strong design feature in LiveCycle Designer is having a table editor where you can easily create table designs with all the necessary background attributes and data formatting.

To create a table in Designer, do the following:

- ▶ Choose Table | Insert table to open the Insert Table dialog.
- ▶ Edit the Columns and Rows field boxes to insert the number of columns/rows you want in your table.
- ▶ Leave the default *Include Header Row in Table* to create a header row. If you want a footer row too, check the box for *Include Footer Row in Table*.
- ▶ Click OK in the Insert Table dialog box.



- ▶ Edit the table using the Border, Font, and Paragraph panels.

# 100

## Adding a Total Field to a Table

You can float fields using subforms in Designer. If you're having trouble getting a subform to float properly so the calculation fields don't move down the page as data are populated in a table, you can add the calculation fields as cells in a table.

To format a table for accommodating a calculation field, do the following:

- ▶ Move the cursor to the left side of a table until you see a left pointing arrow. Click the select the row.
- ▶ Choose Table | Insert | Row Below to insert another row.
- ▶ Drag the cursor through all the cells that won't have a calculation to select them.
- ▶ Open a context menu and choose Merge Cells.
- ▶ Click the cell that requires a formula and drag a Numeric Field to the cell from the Standard Library panel.
- ▶ Name and format the field.
- ▶ Open the Script Editor (CTRL + Shift + 5) and choose Calculate from the Show drop down menu.
- ▶ Type the formula for the calculation in the Script Editor.

| Item         | Price | Qty | Total      |
|--------------|-------|-----|------------|
| Merged Cells |       |     | Calc Field |

## Use the Resources Available to You

For those users who are having difficulty getting started with LiveCycle Designer, there are a number of resources that can help you learn the program and become proficient.

Check out some of these resources:

- ▶ Samples in the Designer 8.0 folder: Open the folder inside the Acrobat 8.0 folder and open the *EN* (for English installations of Designer). Open the Samples folder and review the many sample files installed with Designer.
- ▶ FormCalc Manual: Open the Designer 8.0 | Documentation | EN folder and you'll find a manual for using FormCalc.
- ▶ Visit the Designer Home: Choose Help | Adobe on the Web. Your default Web browser opens and takes you to the LiveCycle Designer Home page
- ▶ Visit the Developer Connection: Log on to: <http://www.adobe.com/devnet/lifecycle/> to find information, tips, and sample files.
- ▶ Visit the Designer Scripting Examples: Log on to: [http://www.adobe.com/devnet/lifecycle/designer\\_scripting\\_samples.html](http://www.adobe.com/devnet/lifecycle/designer_scripting_samples.html) to find sample files.
- ▶ Visit Thom Parker's JavaScript Corner: Log on to: [http://www.acrobatusers.com/tech\\_corners/javascript\\_corner/index.php](http://www.acrobatusers.com/tech_corners/javascript_corner/index.php) to find information, tips, and techniques on Acrobat and LiveCycle Designer JavaScripting.
- ▶ Visit Carl Young's AcroViews: Log on to: <http://www.acrobatusers.com/blogs/pdftrainer/> to find information, tips, and techniques on working with LiveCycle Designer Forms.
- ▶ ted's blog: Log on to: <http://www.acrobatusers.com/blogs/tedpadova/> for tips on using Acrobat and LiveCycle Designer.

# 101 FORMS eTIPS

contact me at: [ted@west.net](mailto:ted@west.net)  
if you have any questions



© not!

