

What's New in Designer 9



This is an online bonus article for Chapter 3 of *Paperless: Real-World Solutions with Adobe Technology*. This article details some important new features in LiveCycle Designer 9. Adobe Acrobat 9 ships with Designer 8.2. All the dynamic form examples in *Paperless* were created with Designer 8.2.

However, if you are using LiveCycle ES2, you have access to Designer 9 (Designer ES2), which offers the same features as 8.2 plus some additional features that you will learn about in this article. Acrobat users should be able to upgrade to this new version of Designer by visiting www.adobe.com.

Working with the New Action Builder

Designer's new Action Builder can be used by nonprogrammers to add automation scripts to a form. Action Builder is a visual tool (**Figure 1**) that you can use to specify custom runtime actions for your forms. After you build an action, Designer automatically writes JavaScript in the form object event that you selected.

An action is a combination of conditions and results. You can have multiple actions in a form, and each action can have one or more conditions and one or more results. You can create an action with one condition and three results by following these steps.

1. Open the `actionBuilderStart.xdp` file in Designer 9 or later.
2. Choose **Tools > Action Builder**. A new Action Builder dialog box appears.
3. Click "Create a new action" in the opening screen.
4. Create a condition by clicking the object link.
5. Select the `highlightRequiredFields` Button object and click OK. Your condition will now be "When button `highlightRequiredFields` is clicked".
6. Create a result by selecting **Set the Background Color of a Field** in the drop-down list (Figure 1).
7. Click the object link and select the `firstName` object.
8. Select yellow in the drop-down color list.
9. Click OK to complete your action. Designer automatically adds custom JavaScript to your button's click event.

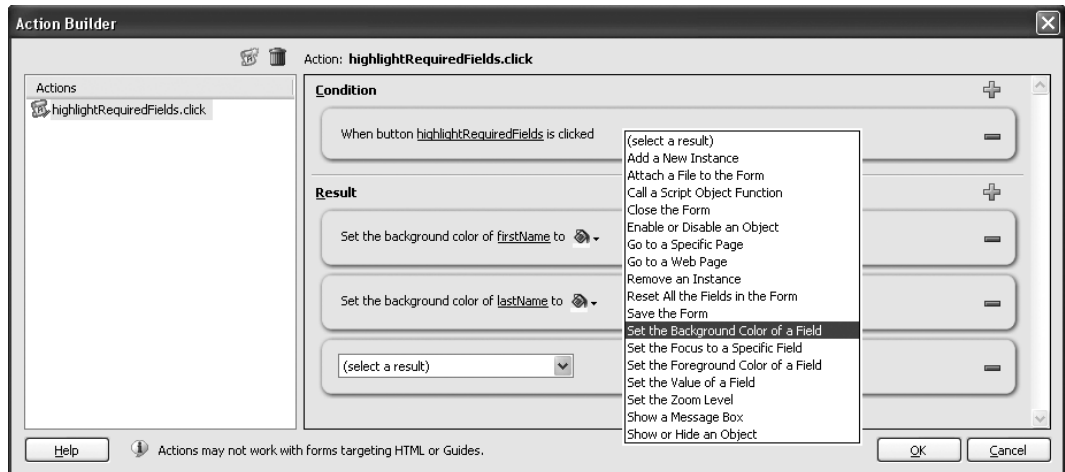


Figure 1 The Action Builder dialog box showing a list of functions you can use to easily add automated functionality to your forms.

10. Select Preview PDF to see the new script in action. When you click the button, the First Name field changes to yellow.
11. You can add additional results to the same condition (Figure 1) by returning to the Action Builder. Choose Tools > Action Builder.
12. Select the highlightRequiredFields.click action.
13. Click the green plus sign on the right to add a new result. Repeat steps 6–8 for the Last Name field and the Email field, and then click OK in the Action Builder dialog box when you are done.
14. Save your file as *myActionBuilder.xdp*.
15. Select Preview PDF and click the button to highlight your fields.

You now have an action with one condition and three results. Action Builder has added the following JavaScript to your button's click event:

```
this.resolveNode("firstName").ui.oneOfChild.border.fill.color.value = "255, 255, 0";
this.resolveNode("lastName").ui.oneOfChild.border.fill.color.value = "255, 255, 0";
this.resolveNode("email").ui.oneOfChild.border.fill.color.value = "255, 255, 0";
```

The list in Figure 1 shows the results that are built into Designer. However, you can expand this list by creating custom results.

Creating custom results

Custom results increase the usefulness of the Action Builder because they enable nonprogrammers to easily access custom JavaScript functions in script objects. Your programming team can create standard JavaScript functions in script objects that your form authoring team can easily add to its forms with the Action Builder. You create custom results by selecting Call a Script Object Function in the drop-down list in the Result panel (Figure 2).

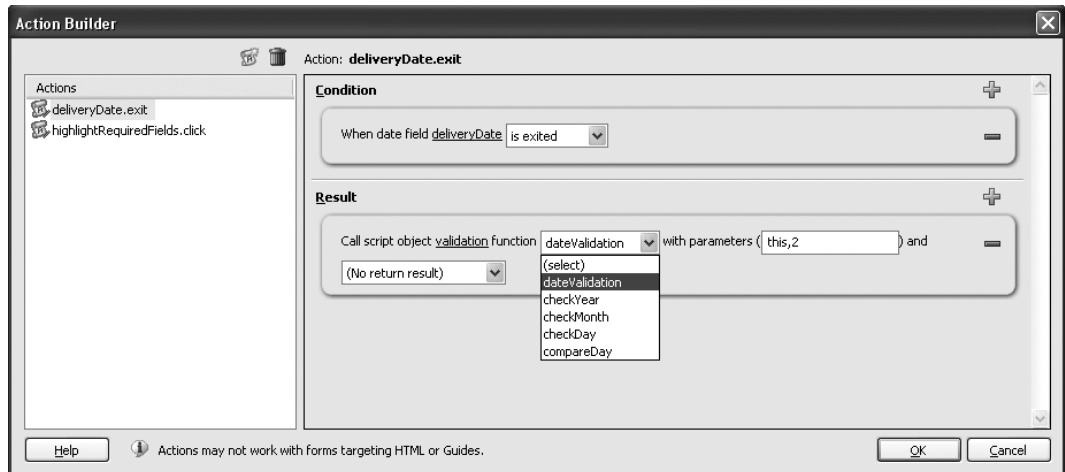


Figure 2 You can create custom results using the Action Builder.

Follow these steps to create a custom result in the Action Builder:

1. Open your myActionBuilder.xdp form in Designer 9 or later.
2. Choose Tools > Action Builder.
3. Click the “Add a new action” icon at the top.
4. Create a condition by clicking the object link.
5. Select the deliveryDate Date/Time field.
6. Select “is exited” from the drop-down list. You now have an action named deliveryDate.exit.
7. Create a result by selecting Call a Script Object Function in the drop-down list.
8. Select “object” to locate the validation script object.
9. Select dateValidation in the drop-down list.
10. Enter *this,2* in the parameters field. These parameters will be passed to the script object. The term *this* refers to the Date/Time field object. The integer 2 ensures that certain JavaScript code is run to check if this date is in the past. The integer parameters are described in the comments section of the script object.


```

      if type = 0 then any date can be entered as long as it is real
      if type = 1 then no future dates can be entered, only real dates
      allowed
      if type = 2 then no past dates can be entered, only real dates allowed
      
```
11. The final drop-down list should display “No return result”. The script does not need to return a result because the validation check and form updating are all contained within the script object.
12. Click OK to create the action.
13. Save your file as *myActionBuilderComplete.xdp*.

14. Select Preview PDF and enter different dates into the Delivery Date field. If you enter a date in the past, a JavaScript warning message prompts you to enter a valid date.

Using the New Form Validation Tab

Designer 9 also includes new validation support for nonprogrammers in the Form Validation tab of the Form Properties dialog box (**Figure 3**). Your users will need Acrobat/Reader 9.1 or later for this feature to work. If you set your target version to an earlier version of Acrobat/Reader, you will see this message in the Warnings tab of the Report palette: “Target version does not support the Color Failed Fields option for form validation.”

If you have Designer 9 and Acrobat/Reader 9.1, you can see this feature in action by following these steps:

1. Open the mySampleSchemaForm.xdp file you saved earlier in the chapter. Alternatively, you can open the file sampleSchemaForm.xdp from the chapter03 folder.
2. Select Preview PDF and enter 11 into the Pants Length field. Acrobat/Reader displays a message box indicating that this entry is invalid. Click OK to close this message box.
3. Switch back to Design View and choose File > Form Properties. The Form Properties dialog box appears.
4. Select the Form Validation tab and select the Color Failed Fields option in the list of options on the left.
5. Select the “Color fields that fail their validations” option in the Settings panel on the right, and then select black for the border color and yellow for the background color.
6. Select Preview PDF and enter 11 into the Pants Length field again. Acrobat/Reader displays the same message box as before, but it also highlights your invalid field in yellow after you click OK in the message box. This is a valuable feature for forms with many fields because the yellow highlight reminds the form filler to update the field.

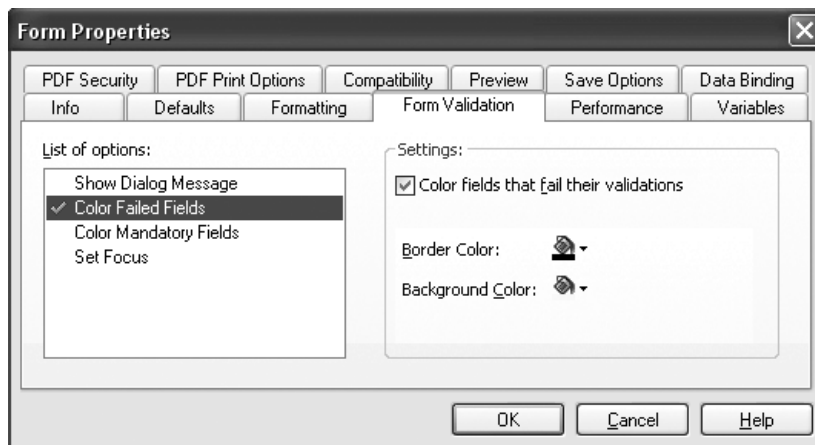


Figure 3 The new Form Validation tab showing the Color Failed Fields option.

Investigating Other Features

A number of other new features and improvements are also in Designer 9. The following is a brief summary:

- **Object Assist dialog box:** The Object Assist dialog box in the Script Editor now contains more details about the XFA methods and attributes, including specific icons for methods, attributes, deprecated methods, and deprecated attributes. This is a valuable enhancement for those who do a lot of scripting in Designer, and it will even prefill your scripts with the parameter types for a method.
- **Default scripting language:** You can now set the default scripting language at the application level for new forms in ES2. Choose the following in Designer 9, Tools > Options > Workspace > Default Language for New Forms.
- **Exported data:** You can now include fields not bound to the schema with your form's exported data.
- **More screen real estate:** You will now have more screen real estate when you use Designer in Adobe LiveCycle Workbench ES2.
- **Populate a List Box:** You can now populate a List Box directly from the clipboard.

Back to the Book

Now that you have learned the technical details about Adobe Acrobat and LiveCycle Designer, go back to the book and read chapters 4 through 6 to see how these technologies are being used in the real world.