

# Introducing the Document Builder in LiveCycle ES2



This is an online bonus article for Chapter 12 of *Paperless: Real-World Solutions with Adobe Technology*. This article details some basic information about the new Document Builder in LiveCycle ES2. As you learned in Chapter 10, LiveCycle Assembler can combine multiple PDF files into a new PDF file at runtime. Assembler relies on a DDX (Document Description XML) file for the specific instructions on how these PDFs should be combined and what the resultant document should look like. Previously, you needed to manually write the DDX file in a text editor, but you can now use the Document Builder, which is a new tool in ES2. The Document Builder is a powerful Workbench perspective that enables you to construct DDX in a visual environment.

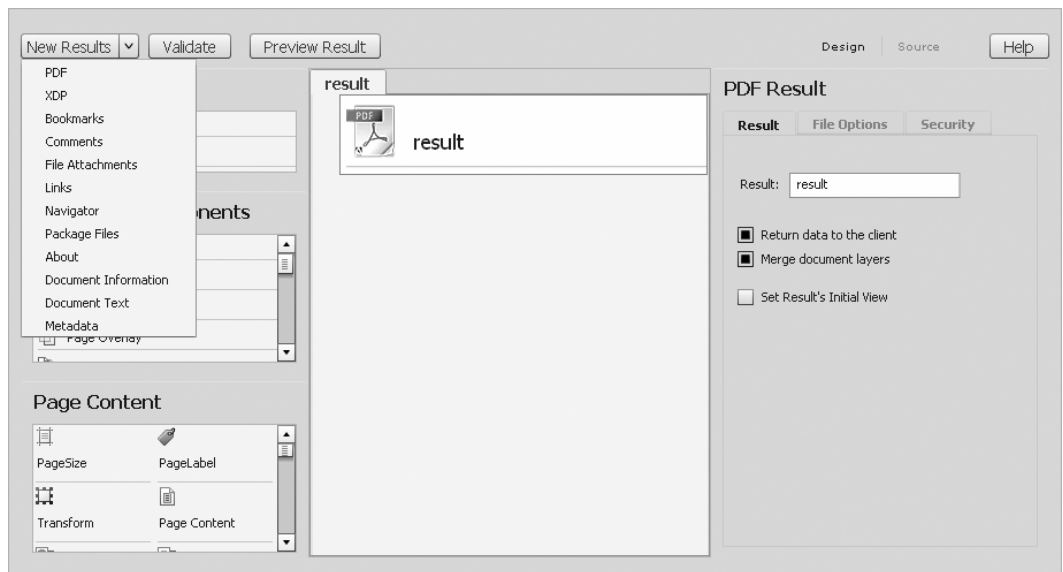
In addition to making DDX construction easier, the Document Builder also expands the type of paperless solutions you can create at runtime: You can now create PDF portfolios on the fly, and you can also stitch together multiple XDP files at runtime. In Chapter 4 you saw an example of a PDF Portfolio, and later in this chapter you will see an example of XDP stitching.

## Combining PDF files with the Document Builder

In this exercise, you will use the Document Builder perspective to add an Assembly Descriptor (DDX file) to the ES2 application you created in the previous example:

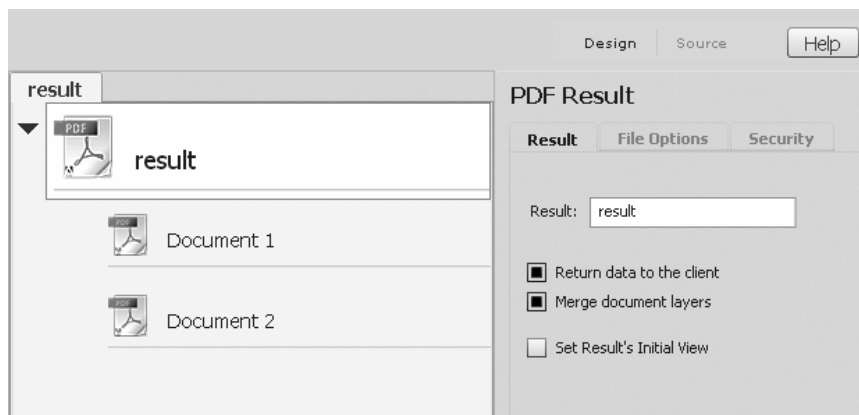
1. Right-click on your MyFirstES2App/1.0 application version in the Applications view of Workbench. Choose New > Folder and create a folder named *DDX*. Click Finish.
2. Drag and drop the *document\_1.pdf* and *document\_2.pdf* files from your *chapter12* folder to the *DDX* folder in your application directory. You will use the same technique that you did previously by dragging and dropping the files from Windows Explorer to your Workbench program.
3. Right-click on the *DDX* folder and choose New > Assembly Descriptor. The New Assembly Descriptor Wizard opens.
4. Enter *assemblePDFs* in the Name field and click Next.
5. This screen has a number of examples that you can explore later to learn more about the Document Builder. For now, choose Empty DDX and click Finish.
6. You may be prompted with a message box concerning the Document Builder perspective. If so, click Yes. The Document Builder perspective opens.

- Click the down arrow to the right of New Results and select a PDF file to be your result. Enter *result* in the Result Name field in the PDF Result panel on the right (**Figure 1**).



**Figure 1** The Document Builder showing the New Results drop-down list on the left and the canvas in the center.

- Drag and drop a PDF from the Sources list on the left to the canvas in the center. A sourcePDF1 entry appears below your result PDF.
- Enter *Document 1* as the Source name in the PDF Source panel.
- Repeat the previous two steps and name the new entry *Document 2*. Your Document Builder perspective should now look like **Figure 2**.



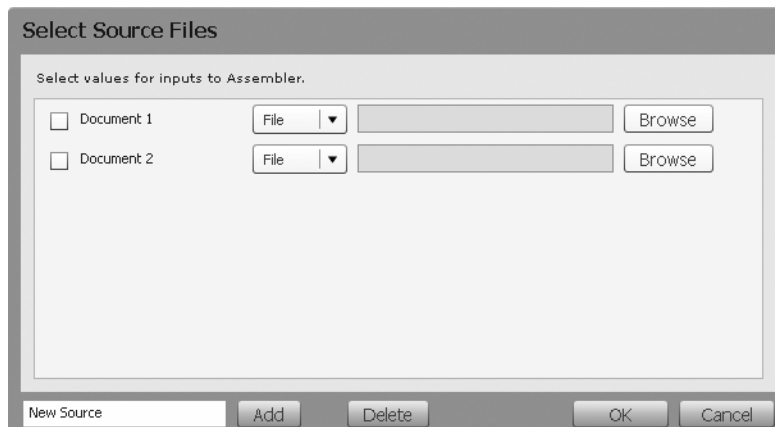
**Figure 2** The Document Builder showing two source PDFs: Document 1 and Document 2.

- Click the Source link in the upper right to see the DDX that the Document Builder has created:

```
<DDX xmlns="http://ns.adobe.com/DDX/1.0/">
  <PDF result="result">
    <PDF source="Document 1"/>
  </PDF>
</DDX>
```

```
<PDF source="Document 2"/>
</PDF>
<?ddx-source-hint name="Document 1"?>
<?ddx-source-hint name="Document 2"?>
</DDX>
```

12. Click Validate on the left. A green check mark indicates this is valid DDX syntax. This is another advantage of the Document Builder; you can now validate DDX at design time to avoid runtime syntax errors.
13. Click Preview Result to see your DDX in action. The Select Source Files dialog box opens (**Figure 3**).



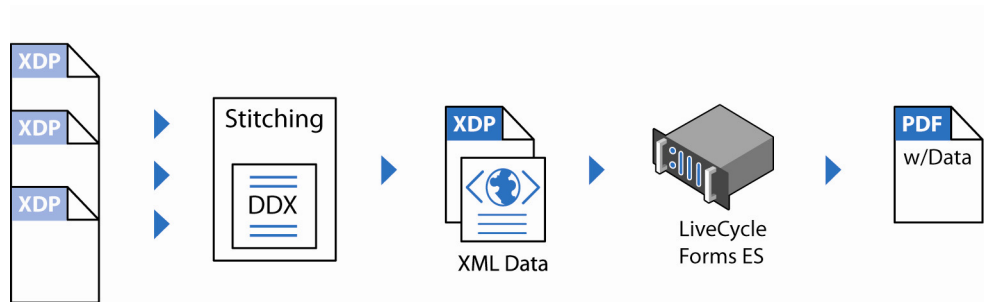
**Figure 3** The Select Source Files dialog box enables you to test your DDX at design time from within the Document Builder perspective.

14. Click the first Browse button and choose the document\_1.pdf file from your application.
15. Click the second Browse button and choose the document\_2.pdf file.
16. Click OK to close the Select Source Files dialog box. The Save Assembly results dialog box appears.
17. Click Open in browser to see the assembled PDF. You now have a three page PDF. Alternatively, you can save the assembled PDF to your local drive.
18. Go back to Workbench and click Finished in the Save Assembly results dialog box. Your new DDX file is saved to the DDX folder of your application.

Many other PDF manipulations are possible with the Document Builder, including an easy to use watermark feature that creates DDX to add a watermark to your assembled PDFs at runtime. Be sure to explore the examples you saw in the Assembly Descriptor Wizard. The DDX documents created by Workbench ES2 must be contained within an ES2 application.

## Combining XDP Files with the Document Builder

The Document Builder also enables you to combine two or more XDP files at runtime to create a composite XDP file—a process known as XDP stitching. This feature is new to Assembler, but some companies have created their own, home-grown tools to combine XDP files on the fly at runtime. Runtime stitching, a valuable tool in many successful paperless solutions, enables a company to maintain separate XDP forms and stitch them together at runtime depending on a user's requirements. For instance, a national financial institution can maintain separate XDP files for each state and combine the appropriate state form with the master national form at runtime. Once the separate files are stitched, the composite XDP can be merged with data and rendered as a PDF by LiveCycle Forms ES/ES2 (Figure 4).



**Figure 4** Multiple XDP files can be combined into one at runtime through a process called XDP stitching.

However, stitching XDP files at runtime creates a couple of challenges. First, you must consider the hierarchies and naming systems used in the source XDP files to make sure these files will not cause conflicts or overwrite each other when they are combined at runtime. Second, you must think about common elements like script objects and formulate a plan for dealing with these common elements when the multiple XDPs are combined.

But the Document Builder does include some important functions for combining XDP files, including the ability to place form fragments at defined insertion points in an XDP file. If combining XDP files at runtime is important for your paperless solution, you should further investigate the new ES2 Document Builder.

## Back to the Book

Now that you have learned about Adobe Acrobat and LiveCycle, make sure to read the last section to learn about a complementary technology: Adobe InDesign Server. Because it comes from the Creative Suite side of Adobe's product line, InDesign Server has graphic capabilities above and beyond what is available in LiveCycle.