PDF HACKS™

100 Industrial-Strength Tips & Tools

O'REILLY®

Sid Steward

HACK #96 Customize Acrobat Using JavaScript

Create custom Acrobat menu items and batch processing scripts.

Acrobat can do most of the things that you need. Yet, there's always something you wish it did a little differently. Acrobat enables you to add custom features using plain-text JavaScripts. These scripts can add menu items to Acrobat's menus or add tailored sequences to Acrobat's batch processing.

Acrobat JavaScript builds on the language core familiar to web developers, but its document object model is completely different from the DOM used by web browsers. Acrobat's JavaScript objects are documented in Technical Note 5186: Acrobat JavaScript Object Specification. Access it online from *http://partners.adobe.com/asn/developer/pdfs/tn/5186AcroJS.pdf*. Another useful document is the *Acrobat JavaScript Scripting Guide* from *http://partners.adobe.com/asn/acrobat/sdk/public/docs/AcroJSGuide.pdf*.

Test Scripts Using the Debugger

The JavaScript Debugger (Acrobat 6 Pro) or Console (Acrobat 5) is the place to test new ideas. Open it by selecting Advanced → JavaScript → Debugger… (Acrobat 6 Pro) or Tools → JavaScript → Console… (Acrobat 5).

Add New Acrobat Features with Startup JavaScripts

When Acrobat starts up, it runs any JavaScripts it finds in either the system-level *JavaScripts* folder or the user-level *JavaScripts* folder. The locations of these folders are given shortly. You might need to create some of these folders if you can't find them.

Use startup JavaScripts to add menu items to Acrobat or to set global JavaScript variables. "Bookmark PDF Pages in Reader" **[Hack #15]** demonstrates how to add a menu item and how to set/query a persistent global variable. Acrobat stores these persistent global variables in a file named *glob.js*. "Pace Your Reading or Present a Slideshow in Acrobat or Reader" **[Hack #10]** shows one use of making Acrobat execute a JavaScript periodically.

These JavaScripts even enable you to add features to the free Reader, although Reader won't perform the more powerful commands. Our various JavaScript hacks all work with Reader. JavaScripts are also platform-independent, so our JavaScript hacks all run on Windows, Mac, and Linux.

Windows startup JavaScripts. For Versions 5 or 6 of Acrobat or Reader, the system-level *JavaScripts* folder is located somewhere such as *C:\Program Files\Adobe\Acrobat 6.0\Acrobat\Javascripts* or *C:\Program Files\Adobe\Acrobat 6.0\Reader\Javascripts*.

For Acrobat or Reader 6, the user-level folder is located somewhere such as *C:\Documents and Settings\Sid Steward\Application Data\Adobe\Acrobat\6.0\JavaScripts*. Both Acrobat and Reader use this one folder.

For Acrobat 5, the user-level folder is located somewhere such as *C:\Documents and Settings\Sid Steward\My Documents\Adobe\Acrobat\JavaScripts*. Both Acrobat and Reader use this one folder.

Sometimes you must create a *JavaScripts* folder, if one does not already exist.

Mac startup JavaScripts. For Acrobat 6, the system-level *JavaScripts* folder is located inside the Acrobat 6 package. Right-click or control-click the Acrobat application icon, and choose Show Package Contents. The system-level *JavaScripts* folder is located at *Contents : MacOS : JavaScripts*.

The Acrobat 6 user-level *JavaScripts* folder is in the user's home folder: *~ : Library : Acrobat User Data : JavaScripts*.

The Acrobat 5 system-level *JavaScripts* folder is located at *: Adobe Acrobat 5.0 : JavaScripts*. The user-level folder is located in the user's home folder: *~ : Documents : Acrobat User Data : JavaScripts*.

Linux startup JavaScripts. For Reader 5, the system-level *JavaScripts* directory is located somewhere such as */usr/local/Acrobat5/Reader/intellinux/plug_ins/JavaScripts*. The user-level directory is located in the user's home directory: *~/.acrobat/JavaScripts*. Create these directories if they don't already exist.

Create Custom Batch Sequence Commands

Acrobat 6 Professional's batch processing **[Hack #93]** uses batch sequences to modify or process collections of PDF documents. Acrobat provides basic commands for creating sequences, such as Insert Pages. Among these commands you will also find Execute JavaScript. Use this command to apply a JavaScript to each PDF in the batch, as shown in Figure 7-10. You can use the *global* object to store the accumulated state of your running batch process, if necessary. Visit *http://www.planetpdf.com/mainpage.asp?webpageid=1511* for some interesting examples and commentary.

Configure Acrobat to use your favorite text editor for editing JavaScripts. Select Edit → Preferences → General… → JavaScript and tell it how to launch your editor.

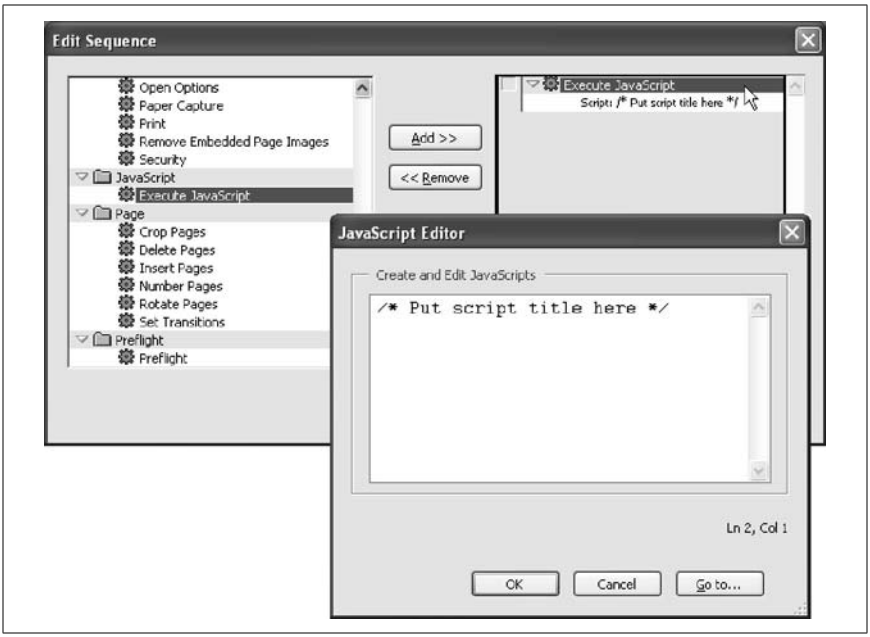

Figure 7-10. Using the Execute JavaScript command to add logic to your batch sequences