

Developer Alert

PDF Rendering in Browsers for Adobe Acrobat 9 and Adobe Reader 9

CONTENTS

Introduction	1
Who Should Read This Document	1
Acrobat Using the Browser HTTP Stack	1
Background	2
Recommended Action	2
Browser Rendering Using In-Process DLLs	2
Background	2
Expected Impact	3
Recommended Action	4
Effect On Debugging	4
PDF Rendering in the Protected Mode Using Internet Explorer on Windows Vista	4
Recommended Action	5

Introduction

This document describes two changes to how PDF documents are rendered in a browser with Adobe® Acrobat® 9 and Adobe Reader® 9:

- In some cases, the browser HTTP stack, including cookies and session information, will be used for HTTP requests instead of using Acrobat or Adobe Reader's HTTP stack.
- In-process DLLs will be used to render PDF documents in the browser.

This document also discusses the implications of Acrobat running as an in-process DLL when Internet Explorer® 7 or later on Microsoft® Windows® Vista® is used in Protected Mode for rendering PDF documents.

Adobe strongly urges developers who may be affected by these changes to test their applications and plug-ins for compatibility with version 9 of Acrobat and Adobe Reader, especially on Vista.

NOTE: *All references to Acrobat in this document refer to both Adobe Acrobat 9 and Adobe Reader 9.*

Who Should Read This Document

- Software developers who use the PDF-in-the-browser context as any part of a server-based HTML/PDF workflow
- Users of Acrobat Interapplication Communication (IAC) mechanisms
- Developers whose software deals with user preferences or communicates information between documents at any level
- PDF workflow developers who use Acrobat in their workflows

Acrobat Using the Browser HTTP Stack

In version 9 and later, Acrobat will use the browser HTTP stack for PDF documents rendered in Internet Explorer, but not in all cases when FireFox is used. This means that for PDF documents being viewed in Internet Explorer, Acrobat will share the browser's HTTP stack, including cookies and session information, for all HTTP requests. For

PDF documents being viewed in Firefox, use of the browser's HTTP stack will be enabled for the following cases:

- JavaScript for Acrobat HTTP POST requests where the `UsePlatformAuth` authentication parameter is set to `true`
- All HTTP requests originating from FormCalc scripts and events in XFA documents

Using the browser context will mean that cookies and session information maintained by the browser will be associated with the PDF and included in HTTP traffic generated by Acrobat.

Background

In the past, HTTP traffic generated by Acrobat from a PDF, other than form submit requests, used the HTTP transport code in Acrobat, which meant using a separate context. While this change will make it easier for applications and plug-ins in the future, it may cause some problems with existing software. In the past, because Acrobat used its own HTTP stack for PDF documents viewed in a browser, some third-party products developed workarounds to associate the PDF-in-the-browser context with the HTML-in-the-browser context. In addition, some third-party products assumed that the PDF-in-the-browser HTTP stack shared cookie and session information with PDF documents that were viewed in Acrobat and not in the browser. This is no longer true for any HTTP requests. The stand-alone context is now in a separate process, so no sharing is possible.

Recommended Action

Software developers who use the PDF-in-the-browser context as any part of a server-based HTML/PDF workflow, and plug-in developers should test their plug-ins and applications with Acrobat 9 as soon as possible. This is especially important to do on Windows Vista due to a related change regarding Protected Mode. (See ["PDF Rendering in the Protected Mode Using Internet Explorer on Windows Vista" on page 4.](#))

Please report all problems using the Acrobat 9 prerelease site.

Expected Impact

This change may affect some developers of server-based software solutions, as well as developers of plug-ins and other types of applications. The changes were implemented primarily in the browser-specific code, and therefore should not affect IAC applications. However, some developers tried to use the browser code path to communicate with, and control, Acrobat, which is an undocumented and unsupported IAC mechanism. In such cases, those developers should take care to test their products with the new version of Acrobat.

Browser Rendering Using In-Process DLLs

Another major change is Acrobat's use of in-process DLLs to render PDF documents in the browser.

Background

Beginning with version 9, multiple instances of the same Acrobat application can be running, each in a different process space. One benefit of this change will be that Acrobat will be more secure in the context of browsing the web with Internet Explorer 7 on Vista. However, developers must be aware

that if they depend on saving state information between instances, they must test carefully with Acrobat 9.

In earlier releases, all PDF documents being viewed in a browser were rendered by a single instance of Acrobat running in a single process space. There was no distinction between one Acrobat instance terminating and all Acrobat instances terminating. Third-party software could use, for example, the plug-in *unload* callback to detect Acrobat shutting down and to save any state information they wanted, using whatever mechanism they designed.

There are many mechanisms by which third-party applications can try to retain (and just as importantly, delete) state information across document openings and closings. Adobe has taken care to try to eliminate preference coordination issues between these multiple instances, but this requires careful testing, especially where third-party preferences are concerned.

A plug-in receives a termination callback for each instance of Acrobat that terminates, and each instance likely has a local cached copy of the state they wish to preserve. For this reason, unless care is taken to coordinate changes between instances, changes could be lost.

Each Internet Explorer process (regardless of the number of windows or tabs or whether there are multiple Internet Explorer instances) will have a distinct instance of Acrobat. Note that starting Internet Explorer from the menu or a short cut will start a new Internet Explorer process. However, creating a new Internet Explorer window from an Internet Explorer menu item will not start a new Internet Explorer process.

Only one Firefox process runs, no matter how many windows and tabs you have open. Even starting Firefox from a menu or short cut will not cause a new Firefox to run. There is, therefore, only one Acrobat or Adobe Reader instance to serve all the Firefox windows and tabs.

The stand-alone instance of Acrobat or Adobe Reader is separate from any browser-based instance. The stand-alone instance of Acrobat or Adobe Reader no longer serves to render any PDF documents in the browser.

Also, plug-ins with workflows in the browser that open local documents and expect to receive the notification `AVDocDidOpen`, for example, will no longer receive it. In fact, the documents will not open at all, which is a security issue. The plug-ins should instead use Weblinks and URLs, or open the documents in another browser window.

Expected Impact

This change will not affect most developers.

Two cases where some issues may be expected include the following:

- Plug-ins that depend on there being only one process for all PDF documents will likely have difficulty because a stand-alone instance of Acrobat will not share process context, including memory, with any browser-based instances.
- PDF documents that explicitly *disclose* themselves to JavaScript scripts running in the context of other documents may not be as visible as they were before. (See the `doc.disclosed` property in the *JavaScript For Acrobat API Reference*.)
- *Reference.*)

Recommended Action

All developers whose software deals with user preferences or communicates information between documents should test with the new release as soon as possible. Please report all problems using the Acrobat and Adobe Reader 9 prerelease site. Also, workflow developers should test and report problems to the developers of modules that have problems.

Effect On Debugging

This issue affects debugging because you can no longer start Acrobat from Visual Studio®, set a break point in your plug-in, and then load a PDF in the browser and expect it to stop at your breakpoint.

There are two solutions:

- You can start Internet Explorer, Firefox, Acrobat, or Adobe Reader (depending on which environment you want to debug in) from Visual Studio.
- You can *attach* to the process that is running the instance you want to debug in, assuming your problem is not *startup* related. If it is, you have to use the first solution.

PDF Rendering in the Protected Mode Using Internet Explorer on Windows Vista

One of the benefits of using the in-proc DLL model in Acrobat 9 is that Acrobat can operate more securely in the context of browsing the Web with Internet Explorer 7 on Vista. By default, Internet Explorer 7 on Vista runs in a *low-rights mode*, which means that the Internet Explorer 7 process, and all in-proc DLLs such as Acrobat 9, are more limited in what they can do.

Because operations such as File > Save A Copy As need more privileges than are available in low-rights mode, Acrobat uses a broker process that allows certain Acrobat APIs to violate the strict low-rights model. The `ASFile` APIs, the `AVPrefs` APIs, and MAPI access through JavaScript or the SendMail HFT, all use a broker process in this way to allow more privileged access than would ordinarily be available in a strict low-rights model. The `AVAppSaveDialog()` function is superseded by `AVAppBeginSave()`, `AVAppEndSave()`, and `AVAppCancelSave()`. These new functions work on Low Rights IE on Vista and take care of the protected mode restrictions internally. The plug-in developer need not worry about which environment is used, and can use these APIs as a replacement for `AVAppSaveDialog`. For more information, see the *Acrobat and PDF Library API Reference*.

If a site is in Internet Explorer's *trusted sites* list, low-rights Internet Explorer will launch a new Internet Explorer process running with normal user rights. This means that if your plug-in on Internet Explorer 7 in Vista shows a non-trusted site and loads another URL from a trusted site, a new instance of Acrobat and thus a new instance of your plug-in will be loaded. If you want to ensure that resources available to Internet Explorer 7 are running in low-rights mode on Vista, you must modify the access permissions for the resource so that the resource is modifiable in low-rights mode. These resources include files, registry settings, shared memory, named pipes, and any object that may be secured by setting their access control lists or through a broker process. If the plug-in uses `ASFile` calls to do all reading and writing, you do not need to consider putting file resources in a low-rights area. Calls to `ASFile` functions will be vectored automatically to `AcroBroker.exe`, the broker process used when Acrobat is running in Protected Mode Internet Explorer.

Also, if the plug-in uses `ASCab` registry functionality, it will automatically have access to `AcroBroker.exe`. When necessary, registry writes that require user permission will automatically be vectored.

Recommended Action

You should test your software and workflows using Internet Explorer 7 on Vista, in the default low-rights mode. If you experience a problem, you can start Internet Explorer 7 in medium-rights mode by doing one of the following:

- Right-clicking the application icon and selecting Run As Administrator
- Double-clicking the Protected Mode: ON box in the status line at the bottom of the Internet Explorer window

If the problem is eliminated under these conditions, you will know that it was related to Internet Explorer running in low-rights mode. You can also turn off User Account Control on Windows Vista to try to eliminate the problem.

Please report all problems using the Acrobat and Adobe Reader 9 prerelease site.

Copyright 2008 Adobe Systems, Incorporated. All rights reserved.

Adobe Systems Incorporated

345 Park Avenue, San Jose, CA 95110-2704 USA

<http://www.adobe.com>

Adobe, the Adobe logo, Acrobat, Adobe LiveCycle, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Mac OS is a trademark of Apple Computer, Inc., registered in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Microsoft, Windows, Windows Vista, and Word are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

28 May 2008