



Acrobat® Online Collaboration: Setup and Administration

November 2006

Adobe® Acrobat® SDK

Version 8.0

© 2006 Adobe Systems Incorporated. All rights reserved.

Adobe® Acrobat® SDK 8.0 Acrobat Online Collaboration: Setup and Administration for Microsoft® Windows®, Mac OS®, Linux®, and UNIX®
Edition 1.0, November 2006

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names and company logos in sample material are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apple and Mac OS are trademarks of Apple Computer, Inc., registered in the United States and other countries.

JavaScript is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Microsoft and Windows are either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

All other trademarks are the property of their respective owners.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

Preface	5
What's in this guide?	5
Who should read this guide?	5
Related documentation	5
1 Introduction	6
Kinds of online collaboration	6
Elements of an online collaboration system	6
Extensibility	7
Client version compatibility	7
Types of comment repositories	7
How FDF files are used in browser-based reviews	8
How JavaScript can be used	8
2 Choosing a Repository Type	9
Assessing requirements and resources	9
Network folder repositories	9
WebDAV repositories	10
SharePoint Services repositories	11
SOAP repositories	11
Repository comparison table	12
Handling comments in network folder and WebDAV repositories	13
3 Configuring Repository Preferences	15
Configuring preferences with InstallShield Tuner	15
Configuring preferences with FDF files	15
Configuring preferences manually	16
Configuring preferences using JavaScript in a document	16
4 Administration Issues	17
User identity and security	17
Data structure	18
PDF open parameters	18
Configuring enterprise service discovery for Acrobat	19
Service name	19
Service type	19
Default DNS domain	20
DNS Service Name	20
PTR DNS record	20
SRV DNS record	20
TXT DNS record	21
Testing the DNS service	21
Using Acrobat Tracker to manage document reviews	22

5	Using a Network Folder Repository	23
	Prerequisites.....	23
	Manually setting a client to use a network folder	23
	Automatically setting a client to use a network folder.....	24
	Network folder issues	25
	Why network paths can be a problem	25
	Network mapping for Windows	25
	Network mapping for Mac OS.....	27
6	Using a SharePoint Repository	28
7	Using a WebDAV Repository.....	29
	Prerequisites.....	29
	Manually setting a client to use a WebDAV server.....	29
	Automatically setting a client to use a WebDAV server	30
	Data storage	31
	Bypassing a proxy server	31
8	Using a SOAP Repository	32
	JavaScript SOAP support	32
	Prerequisites.....	32
	Manually configuring a client to use a SOAP repository.....	33
	Automatically setting a client to use a SOAP repository.....	33
	XML serialization and deserialization.....	34
9	Designing a Collaboration Interface	35
	Online collaboration interface design	35
	Questions to consider.....	35
10	Troubleshooting	37
	WebDAV repositories.....	37
	All repository types	37
	Network problems	39
	Client/server configuration and permissions.....	40
	Index	41

Preface

Adobe® Acrobat® offers a variety of solutions for online collaboration. Because there are several types of comment repositories from which to choose, it is important that you choose the best one for your needs.

What's in this guide?

This document contains information that will help you decide which comment repository to use, and explains how to set up, configure, and administer that repository type.

Who should read this guide?

This document is for system administrators and support personnel who are configuring and supporting an Acrobat online collaboration system. It also provides useful information for application developers and solution providers who use Acrobat as the basis for a custom collaboration system.

Related documentation

For information about	See
Using JavaScript™ to develop and enhance standard workflows in Acrobat and Adobe Reader®.	<i>Developing Acrobat Applications Using JavaScript</i>
Detailed descriptions of JavaScript APIs for developing and enhancing workflows in Acrobat and Adobe Reader.	<i>JavaScript for Acrobat API Reference</i>
Using RSS to track remote resources in an occasionally-connected environment.	<i>Acrobat Tracker</i>
A detailed description of the Adobe PDF file format.	<i>PDF Reference</i>
Detailed descriptions of the APIs for Acrobat and Adobe Reader plug-ins, as well as for PDF Library applications.	<i>Acrobat and PDF Library API Reference</i>

This chapter summarizes the kinds of online collaboration, the types of comment repositories, and the process of uploading, storing, and retrieving comments.

Kinds of online collaboration

There are two kinds of online collaboration:

Shared reviews: The shared review feature is the preferred approach to online collaboration. It uses the latest technologies to synchronize comments and has features to simplify commenting for users. For a shared review, the initiator and reviewers require Acrobat 8 or later.

In shared reviews, reviewers publish their comments to a remote server, yet retain a copy of all comments both in a local cache and within any locally saved copy of the PDF file. Comments are synchronized between the locally saved PDF document and the local cache when the file is first opened in Acrobat, and thereafter comments are synchronized between the local cache of each reviewer and the remote server asynchronously even if Acrobat is not running. This allows reviewers to work disconnected from the review server. A shared PDF document contains the names of all of the invited reviewers, the remote location of the review comments, and information about the initiator.

This guide describes setting up the comment repositories that a shared review can use. For all other information on shared reviews, see the Acrobat Help.

Browser-based reviews: Users can upload a PDF file to a server for a browser-based review. In this kind of review, a remote location for a comment repository is specified in each client. An email message is used to distribute an FDF file to invited reviewers. For more information on FDF files, see [“How FDF files are used in browser-based reviews” on page 8](#)

After a reviewer uploads comments, they become visible to other reviewers. Users who cannot remain connected to the Internet can work offline, then upload their comments later when they reconnect. For a browser-based review, the initiator and all reviewers require Acrobat Standard or Professional 6 or higher.

This guide describes setting up comment repositories for a browser-based review, explains how comment files are handled by the online collaboration system, and discusses aspects related to creating a custom system for browser-based reviews.

You can also collaborate with email-based reviews and ad hoc reviews, but those kinds of collaboration do not use a comment repository. For information on those types of reviews, see the Acrobat Help.

Elements of an online collaboration system

A complete, integrated online collaboration review system includes the following main elements:

- Adobe PDF documents created from source documents authored in various word processing, page layout, and other applications.
- Clients running Acrobat or Adobe Reader (if enabled). For browser-based reviews, the latest browser plug-in should be used. The commenting and reviewing features of Acrobat allow users to invite reviewers, send reminders, view and add comments, and work offline when necessary.
- A central, online comment repository for collecting and storing reviewer comments. For more information, see [“Types of comment repositories” on page 7](#).

Optionally, an online collaboration review system could have a custom user interface to the review and comment workflow. This could include automated routing and notification, implemented by a solution developer using server-side applications. For more information, see [“Designing a Collaboration Interface” on page 35](#).

Extensibility

You can extend the existing online collaboration features in several ways. For example:

- Integrate the online collaboration features into a comprehensive document or workflow management system.
- Write JavaScript code to customize your review process and comment handling.
- Use JavaScript to create new types of comments.
- Customize the manner in which Acrobat handles comments through the use of the JavaScript SOAP object.

Client version compatibility

To ensure that all reviewers can use the same set of comment types, all collaborative participants should use the most recent version of Acrobat or Adobe Reader. This helps to eliminate the possibility that a reviewer using an earlier version of Acrobat or Adobe Reader would not be able to see and respond to comments created using a later version.

Types of comment repositories

A comment repository stores the comment and review items added to PDF documents. When considering the repository types you can use, be aware that you are not limited to a single repository type for every division of your organization. For example, an engineering department might choose a WebDAV repository to allow outside contractors to participate, while the sales and legal departments might both use a network folder repository.

The following types of comment repositories can be used:

Network folder: This type of repository can be used with shared reviews and browser-based reviews. For more information, see [“Using a Network Folder Repository” on page 23](#).

SharePoint: A Microsoft Windows SharePoint Services document workspace holds the comments. This type of repository can be used only with shared reviews. For more information, see [“Using a SharePoint Repository” on page 28](#).

WebDAV: Comments are stored on a web server running WebDAV. WebDAV is a set of extensions to the HTTP 1.1 protocol and allows remote users to cooperatively edit and manage files on web servers via the Internet. This type of repository can be used with shared reviews and browser-based reviews. For more information, see [“Using a WebDAV Repository” on page 29](#).

SOAP: The Acrobat SOAP plug-in allows for the exchange of messages with a remote server. A SOAP-based solution is implemented using JavaScript and a server side HTTP service to manage the reading and writing of the annotation data, and in most cases requires custom development. For more information, see [“Using a SOAP Repository” on page 32](#). This type of repository can be used only with browser-based reviews.

Note: Regardless of the type of repository, all reviewers must have read/write access to the location of the comments.

Each of the standard repository types is discussed in [“Choosing a Repository Type” on page 9](#).

How FDF files are used in browser-based reviews

FDF is an Acrobat format used to store form data and other information from a PDF file. FDF files are used in browser-based reviews for storing comments. An FDF file is created when comments are uploaded. The file contains the reviewer's comments and a reference to the reviewed PDF file. Consequently, the FDF file is typically more compact than the corresponding PDF file, and does not need to be stored under the same parent directory or even on the same machine.

FDF files can also contain JavaScript code that configures the server preferences. For example, a user can navigate to an FDF file on a server and double-click on it. Acrobat then executes the JavaScript code in the file, which configures the server preferences for the user.

For more information about FDF, see the *PDF Reference*.

How JavaScript can be used

Acrobat allows you to use JavaScript to manipulate PDF files, respond to user actions, control the behavior of a document, and do a variety of other actions.

When a PDF or FDF file is opened, Acrobat executes any authorized embedded JavaScript code. For online collaboration, JavaScript code can perform actions such as these:

Set the comment repository preference: For browser-based reviews, the type of comment repository and its location needs to be specified. Those preferences can be set manually or by JavaScript code in an FDF file.

Interface to a web service: JavaScript provides objects to interact with a web service through SOAP. A developer can choose which operations are performed by a given remote procedure, so the JavaScript SOAP support allows for a customized collaboration workflow.

For more information on JavaScript use, see [“Configuring preferences using JavaScript in a document” on page 16](#). For complete information on JavaScript, see the *JavaScript for Acrobat API Reference* and *Developing Acrobat Applications Using JavaScript*.

2

Choosing a Repository Type

This chapter describes the principal features and advantages of each type of repository. For a quick comparison according to such criteria as ease of setup, support for external users, and security, see the table [“Comparison of repository and protocol types” on page 12](#).

Assessing requirements and resources

To create a viable comment repository solution, you must first identify those departments within the organization that will be participating in the document review and commenting process. Each of the repository types has requirements for the operating system and other software needed on the client machines (that of the author and those of the reviewers), as well as the servers.

Consider the following list of questions to help determine the best repository type:

- Are users in close proximity or are they spread over a wide geographical area? Will users outside the organization’s firewall be involved in document reviews?
- What operating systems and software are installed on the users’ machines? Can they be standardized?
- What kind of security is needed for the documents?
- Do you have in-house database expertise?
- Is there a server available for use as the collaboration server?
- Do you have control over the software versions in use by employees and outside reviewers?
- What are the budget constraints?

Your organization may use any combination of repository types. However, for a given PDF document, you should only use one repository type. This helps to ensure that the author and reviewers will be able to see each others’ comments.

Network folder repositories

For the network folder repository type, both the document to be reviewed and the resulting comments can be stored in a network folder.

Since Acrobat uses a hashing algorithm to name the comment store folder to be used for a document located in a network folder, only one platform, either Windows or Mac OS, should be used for all clients when using a network folder repository. This helps to ensure that users will be able to view comments from other users. For more information, see [“Network folder issues” on page 25](#).

While a network folder repository is easy to set up, it is only suited for internal users. If you want to provide access to a network folder to external users, the only solution is to set up an exposed server to which they can connect.

For browser-based reviews, the network folder repository handles comments similar to a WebDAV repository. For more information, see [“Handling comments in network folder and WebDAV repositories” on page 13](#).

The advantages of using a network folder repository include:

- A network folder is the simplest and least expensive comment repository.
- It does not require any special third-party software or databases. Only a file server or web server is needed.

Some disadvantages include:

- If a network folder is used, all clients should use the same platform. If multiple platforms are used, a web server is required.
- Remote users can be difficult to support due to security concerns.

All reviewers must have write access to the parent folder that stores the document review comments. User access to documents and folders can be restricted by modifying their protections.

WebDAV repositories

A WebDAV server can be used for shared reviews or browser-based reviews. For browser-based reviews, Acrobat uses WebDAV commands for adding FDF comment file information to the server collaboration store in a way similar to a network folder repository. For more information, see [“Handling comments in network folder and WebDAV repositories” on page 13](#).

Note: With Microsoft Windows 2000, Acrobat uses a Microsoft COM object to access the WebDAV server. This COM object is not case-sensitive and does not distinguish between such user names as *JPUBLIC* and *jpublic*.

While the client configuration for WebDAV is simple, it can be problematic because of IT configuration and security concerns. If the proxy server supports the HTTP 1.1 protocol and the port is specified, remote users will have access through a company firewall. If HTTP 1.1 is not supported, a user can bypass the firewall by using an exception list for the proxy (see [“Bypassing a proxy server” on page 31](#)). As with most repository types, a WebDAV repository can be set up in a secure manner. HTTPS can be used, and both authentication and digest authentication is supported. However, some firewalls and proxy servers do not support WebDAV. Ensure that all reviewers have read and write access to the WebDAV folder that holds the comments.

A significant advantage of WebDAV is that it supports remote users. WebDAV requires proper setup of a directory that will serve as the parent directory for all document review comments. WebDAV has the fastest response time of the repository types that allow remote user access, except when there is a large number of users.

For browser-based reviews, WebDAV requires that the entire FDF file be transferred each time comments are added, updated, or removed. In addition, one connection per user is established when downloading. Thus, file size and the number of users should both be considerations when determining the likely performance of this type of repository.

A WebDAV solution offers you flexibility in tailoring a comment repository interface that closely supports the business practices of the company.

SharePoint Services repositories

You can use a Microsoft Windows SharePoint Services document workspace as a comment repository for shared reviews. If your organization is already using SharePoint Services, you can leverage this resource by choosing it for the repository. Such a repository is easy to set up and configure, and is included with Windows Server 2003.

All reviewers must have read and write access to the workspace on the Windows server that is running SharePoint Services.

For more information on SharePoint Services, see <http://www.microsoft.com/windowsserver2003/technologies/sharepoint/default.mspx>.

SOAP repositories

The Acrobat SOAP support can be the basis for creating a customized collaboration workflow for browser-based reviews. JavaScript provides objects with which remote procedure calls can be sent from clients to a server, but there is no user interface to help the clients handle such operations as comment uploading.

There are several advantages to using SOAP as the basis for a custom collaboration system:

- All transactions are based on XML, ensuring greater interoperability with XML-compliant applications.
- SOAP repositories can make it easier to support remote users because SOAP's reliance on HTTP makes it easier for SOAP envelopes to pass through firewalls.
- Single comments or other types of updates can be added to the comments store without having to transmit the entire FDF file each time. This can save time and reduce network traffic, especially for large documents, providing a comparative advantage over the WebDAV approach.
- You can use a wide variety of data store types for saving the annotation data submitted to the server.
- Only one HTTP connection is required to retrieve all of the comments. This may be an important performance consideration for a high latency connection.
- SOAP-based development is well supported because of the widespread availability of standard tools, and offers the advantages of independence from platforms and development environments. In addition, the JavaScript SOAP support provides several convenient methods that simplify development tasks by eliminating the need to work directly with XML.
- SOAP uses only the HTTP POST verb and is therefore more likely to work through a firewall than WebDAV, which requires other HTTP verbs.

In comparison with WebDAV, SOAP repositories typically use more space on the server, and the serialization and deserialization process associated with XML parsing can consume more processing cycles than other methods. For more information on the serialization and deserialization process, see ["Using a SOAP Repository" on page 32](#).

There are significant advantages to developing a custom SOAP repository that addresses your specific requirements. For example, you can design the ideal collaboration user interface along with custom routing and approval processes. For more information, see ["Designing a Collaboration Interface" on page 35](#). This would, however, require more development effort than setting up a standard collaboration solution.

Repository comparison table

The following table provides an overview of the features, costs, and benefits of the repository types.

Comparison of repository and protocol types

	Network folder	WebDAV	SharePoint Services	SOAP
Shared reviews	Yes	Yes	Yes	No
Browser-based reviews	Yes	Yes	No	Yes
Third party software needed?	No	See footnote	Microsoft Windows SharePoint Services must be configured and running.	Yes. Many open source options are available.
Client OS	For shared reviews, Windows or Mac OS. For browser-based reviews, one platform is recommended.	Windows or Mac OS	Windows Mac OS clients can participate in reviews but cannot initiate reviews.	Windows or Mac OS
Other Client Software	No	No	Microsoft Office 2003 or Microsoft Windows XP	Browser
Supports remote users?	Not well	Yes, if the web proxy supports it	Depends if the website is internal or external to the firewall	Yes, depending upon internal and external access needs and firewall/proxy issues
Performance	Good	Fastest if only a few users. For a large number, having one HTTP connection per user slows performance.	Slower than WebDAV	Good
Security	Good	Good	Depends on implementation	Good if set up correctly
Setup	Easy	Depends on network size and security (firewall/proxy server)	Easy	Requires custom development

	Network folder	WebDAV	SharePoint Services	SOAP
Web server needed?	No	Yes	Yes	Yes
Web server OS	If used, Windows 2000 SP4 or higher	Windows 2000 SP4 or Windows XP; most versions of UNIX	Windows Server 2003	Supported on most operating systems
Web server software	Any	Apache 1.3.14 and mod_dav, Apache 2.0 or higher, or Microsoft IIS 5.0 or higher	Microsoft IIS 6	Only if you prefer to use a web server

Handling comments in network folder and WebDAV repositories

For browser-based reviews, the network folder and WebDAV comment repositories both handle comments in a similar way. This example shows the similarities and differences between these types of repositories. Understanding this process may help you to choose the type of repository to use for browser-based reviews.

Suppose an author has initiated a review of a PDF document, called `Brochure.pdf`. The file is posted to a web server for review, and the reviewers have been notified to use this URL:

```
http://yosemite.example.com/webdav/Brochure.pdf
```

The company's IT department has already configured everyone's copy of Acrobat to use a WebDAV repository and to use the following server location to store comments:

```
http://bryce.example.com/repository/
```

No one has yet reviewed the document.

A reviewer named Michelle Barnes is the first to access the document for review. She marks up the PDF file and clicks Send Comments. Acrobat uses Michelle's Windows log-in name, which is `mbarnes`, to construct the name of the FDF file and exports these comments into `mbarnes.fdf`.

Next, Acrobat uploads Michelle's comment file to the right location. Acrobat applies an MD5 hash algorithm to the fully qualified URL of the document:

```
http://yosemite.example.com/webdav/Brochure.pdf
```

which results in the value:

```
jY_d0VPgIZG6_jUknGUJbzA
```

This hash algorithm ensures that the resulting folder name is legitimate and manageable, because a URL may contain characters that cannot be used in a folder name. This hashed folder name is where Acrobat will upload all FDF files that contain comments for the file `Brochure.pdf`. Since this folder does not exist, Acrobat creates it under the following URL and then uploads `mbarnes.fdf`:

```
http://bryce.example.com/repository/
```

For more information on how Acrobat handles a URL, see ["Data structure" on page 18](#).

Later, a second reviewer, Jim Davidson, accesses the document for review. Acrobat retrieves the following URL and checks the comment folder for any FDF files:

```
http://yosemite.example.com/webdav/Brochure.pdf
```

Acrobat determines that there is one FDF file relevant to that document, `mbarnes.fdf`, in the comment folder. Acrobat imports the comments into `Brochure.pdf` and displays the result. Jim now sees the PDF file with Michelle's comments.

When Jim finishes his review, he publishes his comments. Acrobat uses Jim's log-in name of `JDavidson` to create the name of the FDF file that will hold his comments. Acrobat exports only Jim's comments into `JDavidson.fdf`. Michelle's comments, and those of all other reviewers, are excluded from `JDavidson.fdf`.

A difference between the network folder and WebDAV repositories is their treatment of log-in names. The network folder repository may be case-sensitive. Consequently, users with identical log-in names that differ only by case, such as `mbarnes` for Michelle Barnes versus `MBarnes` for Martin Barnes, can comment on the same document in a network folder repository since their resulting comment FDF files will be different ("`mbarnes.fdf`" versus "`MBarnes.fdf`"). This is not the situation with WebDAV, for which all reviewers must have unique log-in names that are not differentiated solely by case. For more information, see ["User identity and security" on page 17](#).

The network folder repository is the only type in which the initiator of a review can provide a full path to a document instead of a URL (such as `\\rainier\marketing\brochures`). However, it is important to note that all reviewers must use the same path, and ideally they would all be on the same platform. Otherwise the MD5-based hash value will be different and reviewer comments will be stored in several different folders. When this occurs, reviewers will not be able to automatically see all review comments made on the document by other reviewers (see ["Why network paths can be a problem" on page 25](#)).

For example, suppose an author sends an email notification to reviewers that the document `ToySpecs.pdf` is located in `\\Rainier\Engineering\Documents`. Reviewer Samantha Wanke has mapped network drive R: to `\\Rainier\Engineering` while reviewer Denny Brown has mapped network drive G: to the same path. Samantha and Denny will never see each other's comments because the different drive mappings result in different hashed values. These problems can be avoided either by having the IT department set up consistent drive mappings on all user machines or by providing a URL to the document location.

These problems also can be avoided by using a shared review because shared reviews work independently of where the document is located.

3

Configuring Repository Preferences

For shared reviews, the initiator configures the repository by using the shared review wizard. Configuration can also be set by using InstallShield Tuner.

For browser-based reviews, you set the configuration preferences for each client in one of the following ways:

InstallShield Tuner: The system administrator can set the installer to configure each user's system for the required preferences when Acrobat is installed.

PDF files: An FDF file containing JavaScript can be used to set the preferences.

Manually: Users can set the preferences in Acrobat individually.

Embedding JavaScript: In this approach, JavaScript code is placed in the PDF file to be reviewed. When a reviewer opens the file, the code is executed and the preferences are set. This approach makes it more difficult to repurpose the PDF file, and is not recommended.

Additional instructions for setting preferences for browser-based reviews are provided in the chapters about each repository type.

Configuring preferences with InstallShield Tuner

The simplest way to ensure that all the users in a large organization have the same repository preferences is to use InstallShield Tuner. With this tool, a system administrator can define what the preferences are before Acrobat is installed on each client, and each client will have the identical preferences.

You can configure the repository preferences, set Synchronizer settings, and more. For more information, see the documentation that comes with InstallShield Tuner.

Configuring preferences with FDF files

FDF files are used by Acrobat to receive information from a server and to fill in form fields. For browser-based reviews, an FDF file specifies the PDF document to which it applies, and can specify the type of comment repository to be used and the location of the comments. The FDF file can also communicate with the JavaScript engine, which is used to automatically set the users' server preferences.

You can create a single FDF file to automatically configure the preferences for all users who open the file in their browsers. Assigning the comment repository in an FDF file precludes the possibility of any user error in setting up the comment repository. It is a flexible approach that allows you to specify that the comment repository applies to all PDF files or to a particular PDF file.

For example, a user can be directed to open an FDF file that has been coded with the preference information. When Acrobat processes this FDF file, it retrieves the associated PDF file, sets the repository preferences, and imports any reviewer comments.

Alternatively, a network administrator can create an FDF file for each repository type, ensuring that the server preferences are automatically set for the user. This is useful for people who frequently comment on multiple documents. For ease of use, the administrator can place the FDF file at a readily accessible location, such as a frequently used web site.

Most authors will not have the desire or expertise to create an FDF file that specifies comment repository preferences for an individual PDF document. Consequently, this approach works best if the repository administrator creates the FDF file when an author adds a document to the repository. Fortunately, this process can be automated using an intelligent front-end, developed by solution implementors, with which authors interact.

Configuring preferences manually

If your organization or department needs only one type of repository for its browser-based reviews, it might be best to configure it manually, since end users will not need to change the settings. The system administrator can assign comment repository settings when configuring a workstation for a new user. This eliminates the possibility of users configuring the repository preferences incorrectly.

Comment repository preferences in Acrobat have global scope. In other words, the preferences apply to every PDF file that a user views. If the comment repository is located on a server that requires a user name and password, the login prompt appears for every PDF file a user opens in the web browser, even for PDF files located on websites outside the company. To address this issue, you may set up the repository preferences on a per-document basis using either FDF files or document-level JavaScript code.

Each user must set the comments server preferences before their collaboration session can begin. This can be done by selecting Edit > Preferences > Reviewing. The server preferences remain in effect until changed.

Configuring preferences using JavaScript in a document

As an alternative to creating an FDF file for browser-based reviews, and to avoid having to reset the server type and location for each collaboration session, the server preferences can be set in document-level JavaScript code which runs whenever the document is opened.

Note: While embedded JavaScript code can be used, it is not recommended because it makes it more difficult to repurpose a PDF document and it causes a security dialog box to appear during the review process.

► To configure JavaScript preferences:

1. Select **Advanced > JavaScript > Document JavaScripts**.
2. Enter a name for your script.
3. Click **Add** and delete the default function script.

For example, for a WebDAV repository, you could insert code similar to the following:

```
// Document level script
this.collabStore = Collab.createAnnotStore({ cName: "DAVFDF", doc: this,
                                             cUser: "janedoe", cSettings: "http://example/Collab/" });
```

Note: The `cSettings` value requires a terminating slash (/) character.

All parameters are optional, although you would typically supply all except for `cUser`, which may be difficult to obtain due to security restrictions on identity information.

The code associates the folder with the WebDAV server type. A device-independent path identifies the folder holding the comments. For more information on device-independent paths, see the *PDF Reference*. For more information on JavaScript support for online collaboration, see the *JavaScript for Acrobat API Reference* and *Developing Acrobat Applications Using JavaScript*.

This chapter describes administration issues such as user identity and security, data structure, PDF open parameters specified in URLs, and automated enterprise service discovery.

User identity and security

Since several users may comment on a single document, it is essential that each of the comments is attributed to the correct person. To ensure that the comments are attributed correctly, Acrobat automatically inserts the name that appears as the Login Name, shown on the Identity panel within the Preferences dialog box, into the title bar of the annotation pop-up note.

For shared reviews, when users open a PDF for review for the first time, they are prompted to confirm or enter their reviewer profile. However, this does not occur with browser-based reviews.

By default on Windows clients, Acrobat uses the Windows login name. On Mac OS clients, Acrobat uses either the login name or the file sharing name, depending on which exists. User names are generally case sensitive, which means that *JPUBLIC* is a different user than *jpublic*. For an exception to this rule, see [“Using a WebDAV Repository” on page 29](#).

Note: If the user has bypassed the Windows login screen, the default user name is “GUEST.” This means that when the user uploads their comments, those comments will be attributed to the anonymous “GUEST” account and not to the user who made the comments. This may cause confusion.

For network folder and WebDAV repository types, when the login name contains something other than letters or numbers, Acrobat substitutes the hexadecimal representation of that character within the login name. For example, comments for a user with the login name of `John Public` would be stored on the server in a file called `John_0020Public.fdf` because the space in the name is translated as a hexadecimal '0x0020'.

Acrobat does not use a checkout or locking mechanism for comment files. Therefore, to prevent confusion and possible loss of information, do not allow multiple users to share the same user name.

Note: On a Windows server hosting a file using Apple File Services, a 31 character limit for a file name applies. In this situation, be aware that users with many spaces or non-alphanumeric characters in their user name may exceed that limit.

Acrobat controls access by allowing comments to be uploaded only with the login name, and only if the login name has permission to create files on the server. In the case of a default login name, the comments would be uploaded as `GUEST.fdf`. If there were more than one person logged in as “GUEST” the comments of each user would be saved to the same file.

Acrobat does not verify that the user has write privileges to either the root or hash directories. If the user does not have write privileges, or the directory does not exist, Acrobat displays an error message stating that the comments could not be saved and uploaded.

Data structure

When a user opens a PDF document for browser-based review, Acrobat reads the URL from the network layer and then uses Domain Name Server (DNS) to translate the name to an IP address. The DNS conversion guarantees that the comments will be stored in the correct repository regardless of whether the users specify the same URL to access the collaboration document. Acrobat then converts the IP address into an MD5 hash in order to create the name for the comments folder. For example:

```
http://www.example.com/docs/document.pdf
```

would be converted to

```
http://192.0.34.166/docs/document.pdf
```

which would subsequently hash into, for example, UAtOJErFgHsL2xbLvPPJgC. More information on MD5 hash can be found in RFC 1321.

Note: You cannot reverse the MD5 hash to obtain the URL.

For browser-based reviews, do not change the location of the collaboration document until the collaboration period has expired, because the folder names are based on the URL of the original document. If the location of the collaboration document is changed and more comments are added, the new comments will be saved to a different location from that used for the initial comments.

When working with multiple versions of a document, name each version differently; otherwise the comments will not align properly. In the example above, the MD5 hash for

```
http://152.30.132.12/docs/document.pdf
```

after the URL is put into a canonical form, is:

```
UAtOJErFgHsL2xbLvPPJgC
```

while the MD5 hash for:

```
http://152.30.132.12/docs/document2.pdf
```

is:

```
1N2AwJKLt6BxpzjHp4_W7B
```

If you were to revise `document.pdf` and place the updated document in the same location without changing the file name to reflect the version, the comments saved under that hash would not correlate to the new document. This could result in such problems as highlighting that exists where there are no longer any words, or line drawings for a diagram that has moved to a different page.

PDF open parameters

You can use PDF open parameters to open a document with a command or URL that specifies what to display, such as a specific page, and how to display it, using characteristics such as scrollbars, bookmarks, and highlighting. You can also set the collaboration store and open or remove items in Tracker. For more information, see the guide *Parameters for Opening PDF Files*.

Configuring enterprise service discovery for Acrobat

Acrobat provides the ability to locate services automatically using DNS Service Discovery. Services that are advertised using Multicast DNS can be located without manual configuration if the service and client are located within the same network subnet. To discover services throughout a corporate network, however, services are located using Unicast DNS queries and the existing DNS infrastructure.

To advertise a service throughout an enterprise by configuring Unicast DNS you must configure the following settings:

- Service name
- Service type
- Default DNS domain
- DNS Service Name
- PTR DNS record
- SRV DNS record
- TXT DNS record

After the service has been configured you should test the settings.

It is assumed that you have expertise working with DNS and BIND.

Service name

Choosing an appropriate name is crucial for advertising the service. The name will be displayed to the user as part of the Acrobat user interface, and should convey sufficient meaning for the users to decide whether it is appropriate for their needs. The name must be unique and should indicate the purpose of the service (for example, "Accounting Collaboration Service"), and possibly its location (for example, "Engineering Collaboration Service"). The name can contain spaces and other special characters. When the service name is entered into DNS, the special characters must be escaped by replacing them with their decimal UTF-8 character code. For example, a space character is expressed as \032. The preceding examples would be entered in DNS as follows:

Accounting Collaboration Service: Accounting\032Collaboration\032Service

Engineering Collaboration Service: Engineering\032Collaboration\032Service

Service type

Acrobat searches for different types of services in different parts of the user interface. The service type should correspond to the service that you intend to publish. The service types and descriptions are as follows:

Service type	Description
<code>_acrobatSRV._tcp</code>	An Acrobat service
<code>_acrobatRMACR._tcp</code>	An Acrobat review, markup, and approval repository

Default DNS domain

Acrobat searches for services in the default DNS domain. On Windows, this can be determined by executing the `ipconfig /all` shell command. The default DNS domain corresponds to the "Connection-specific DNS Suffix" value.

DNS Service Name

The DNS Service Name is the name of the service as it will appear in the DNS entries, and can be created from the Service Name, Service Type, and Default DNS domain. The DNS Service Name has the following form:

```
<Service Name>.<Service Type>.<Default DNS Domain>.
```

If the Service Name contains special (non-alphanumeric) characters, those characters must be escaped. For example, a space must be replaced with `\032`. The period (.) appearing after the Default DNS Domain, must be included. For example, if the Service Name is "Example Service Name" for an Acrobat service, and the Default DNS Domain is "corp.example.com", the DNS Service Name would appear as follows:

```
Example\032Service\032Name._acrobatSRV._tcp.corp.example.com.
```

PTR DNS record

The first DNS entry needed to advertise the service is a PTR record that maps the Service Type to a particular service instance. In BIND the entry would appear as follows:

```
<Service Type>.<Default DNS Domain>. IN PTR <Service Name>
```

The period (.) appearing after the Default DNS Domain must be included. Using the same example, the entry would appear as follows:

```
_acrobatSRV._tcp.corp.example.com. IN PTR Example\032Service\032Name
```

SRV DNS record

The SRV record allows the Service Name to be bound to a specific IP address and port corresponding to the service. In BIND the entry would appear as follows:

```
<Service Name> IN SRV 0 0 <port #> <Service Target>
```

If the service above were running on port 8080 of the machine `exampleService`, the entry would appear as follows:

```
Example\032Service\032Name IN SRV 0 0 8080 exampleService.corp.example.com
```

TXT DNS record

The TXT record allows the Service Name to be bound to a set of named parameters that are specific to the service you are publishing, and are described in the following table.

Service type	TXT parameters
<code>_acrobatSRV._tcp</code>	<p>txtvers: The version number of the service. This is 1 for Acrobat 8.0.</p> <p>path: The path portion of the URL for the service. For example, if the URL is: <code>http://example.org/services/ExampleService</code> then the path parameter would be set to: <code>services/ExampleService</code>.</p> <p>feedType: The Service Type (either "RSS" or "JS"). "RSS" indicates that it is an RSS feed to be deployed in Tracker, and "JS" indicates that it is an Initiation Wizard.</p>
<code>_acrobatRMACR._tcp</code>	<p>txtvers: The version number of the service. This is 1 for Acrobat 8.0.</p> <p>setting: The collaboration store setting. This depends on the type of repository being used.</p> <p>type: The collaboration service type. The possible values are "DAVDFD" (WebDAV), "FSFDF" (Network Folder), and "CONFIG" (Custom Server Type).</p>

In BIND the entry would appear as follows:

```
<Service Name> IN TXT "<p1 name>=<p1 value>" "<p2 name>=<p2 value>" ...  
"<pN name> = <pN value>"
```

For example, if the service above is an Acrobat service with a feedType of RSS and a path of RSS/example.rss, then the entry would appear as follows:

```
Example\032Service\032Name IN TXT "txtvers=1" "feedType=RSS"  
"path=RSS/example.rss"
```

Testing the DNS service

The service should now be advertised using DNS. To test that it has been advertised properly, you can use `nslookup`.

To find the services for a specific service type, use the following syntax:

```
nslookup -q=ptr <Service Type>.<Default DNS Domain>
```

For example, to search for an Acrobat service in the domain `corp.example.com`, use the following syntax:

```
nslookup -q=ptr _acrobatSRV._tcp.corp.example.com
```

This will return the set of service names that are advertised as Acrobat services. To obtain the SRV and TXT records for a specific service name, use the following syntax:

```
nslookup -q=any <Service Name>.<Default DNS Domain>
```

For example, to obtain the SRV and TXT records for a service named "My Example Service", use the following syntax:

```
nslookup -q=any My\032\Example\032Service.corp.example.com
```

Using Acrobat Tracker to manage document reviews

Acrobat Tracker can be used to monitor PDF document reviews by publishing information using RSS 2.0, which is an XML-based format used for describing lists of items. It can also be used to produce a visual client for a web service application.

The Tracker URL API can be used to manage subscriptions through the use of application integration. This is accomplished with a specialized `acrobat:` notation, which uses XML extensions to customize the user interface. This syntax can be used to create links that add, update, or remove subscriptions to document reviews, open a PDF document in the browser, or display the Tracker window, as shown in the following examples:

Action	URL
Show Tracker	<code>acrobat:Inbox?show</code>
Add a feed to Tracker	<code>acrobat:Inbox?addFeed=http://www.example.org/rss/example_rss.xml</code>
Refresh the same feed in Tracker	<code>acrobat:Inbox?refresh=http://www.example.org/rss/example_rss.xml</code>
Select the same feed in Tracker	<code>acrobat:Inbox?select=http://www.example.org/rss/example_rss.xml</code>
Convert the same feed to PDF. If it is not currently subscribed, it is converted without adding it to Tracker.	<code>acrobat:Inbox?convert=http://www.example.org/rss/example_rss.xml</code>

For more information, see the *Acrobat Tracker* guide.

This chapter discusses network folder comment repositories and how to configure a client to use one. The network folder repository is a simple mechanism for collaborative review using a folder on a network. This is the easiest of the comment repository types to use because it does not require a web server or any coding.

For browser-based reviews using a network folder repository, the collaboration document is placed on the network and the reviewers are notified. As each reviewer opens and comments on the document, Acrobat creates an MD5 hash of the document URL, and the comments are saved in the hashed sub-folder as an individual FDF file. The PDF file and its accompanying FDF files do not need to be stored under the same parent directory or even on the same machine.

For shared reviews, comments are stored within the PDF document and in XML files on the comment server.

Prerequisites

The following are requirements for the comment repository server:

- A network folder must be chosen to store the comments and it must be accessible and allow write access to all reviewers.
- A local drive can be used to store the comments, but then comments can be created and viewed by only one user unless access to that drive is shared with the network.

The following are requirements for each client:

- All users must have either Windows or Mac OS systems.

Manually setting a client to use a network folder

For shared review, you set the client to use a network folder repository when initiating a review with the shared review wizard.

For browser-based reviews, the repository can be set on each client manually as follows.

► **To manually set the preferences:**

1. In Acrobat, click **Edit > Preferences** and select **Reviewing**.
2. Change the **Server Type** to Network Folder.
3. Click **Browse** and select the network folder where the comments are to be stored.

For example, suppose that users John Public and Jane Doe will be commenting on a document called `http://www.example.com/docs/document.pdf`, and that the comments will be held in a network folder called `F:\Example\collab\annots\`.

When John Public and Jane Doe create comments and upload them to the server, Acrobat creates a hash folder and saves their comments as:

```
F:\Example\collab\annots\48dsn30fhbfw\jpublic.fdf
```

and:

```
F:\Example\collab\annots\48dsn30fhbfw\jdoe.fdf
```

Although this example uses the letter name of the drive, Acrobat also supports UNC paths. The equivalent UNC path would appear as follows:

```
\\diskfarm\collab\annots\48dsn30fhbfw
```

Automatically setting a client to use a network folder

For browser-based reviews, the repository preference can be set automatically by using an FDF file containing JavaScript code.

The following example script sets the repository to `\\fileserver\shared\Collab`, and the store type for a network folder repository as `FSFDF`. The `Collab.setStore` method's second argument is a device-independent path expressed with UNC path representation. For more information on device-independent paths, see the *PDF Reference*.

```
%FDF-1.2
%âãÏÓ
1 0 obj
<<
  /FDF
  <<
    /F (http://reviewserver/pdfs/settingsupdated.pdf)
    /JavaScript
    <<
      /After 2 0 R
    >>
  >>
endobj
2 0 obj
<<
>>
stream
  Collab.defaultStore = "FSFDF";
  Collab.setStoreSettings("FSFDF", "\\fileserver\shared\Collab/");
endstream
endobj
trailer
<<
  /Root 1 0 R
>>
%%EOF
```

In this example, `http://reviewserver/pdfs/settingsupdated.pdf` is the PDF document that opens in the browser after the collaboration settings have been set. To adapt this script to your needs, modify the PDF file name and the store settings.

Network folder issues

For browser-based reviews, specifying a network path to the location where a document is placed for review can be problematic because of the way networks function. Windows and Mac OS systems map network drives in different ways, and there are variations within each environment. You can solve the problem through the use of a web server.

Why network paths can be a problem

On Windows, a file name could appear as follows:

```
\\mainserver\users\jsmith\example.pdf
```

or:

```
D:\jsmith\example.pdf
```

On a Mac OS system, the file might be specified in different ways depending on the operating system used. The syntax in the following example is used in Mac OS 9.0, but would appear differently with Mac OS 10:

```
Server Zone:mainserver:Users:jsmith:example.pdf
```

Since the syntax used in each case is different, Acrobat creates different hashed-name folders in the comment repository.

A successful online collaboration workflow depends on a single name that is accessible from any platform. In such situations, you should use HTTP-based URLs, which are not case-sensitive and rarely contain platform-dependent syntax. If you must use a network folder to hold the PDF, then it will only work consistently for one platform, and you may need to take steps to ensure that there are no variations within that platform that could result in different file name formats.

Using a web server as the location of the document to be reviewed can help reduce hashing problems associated with using a network folder as a comment repository.

Network mapping for Windows

There are four ways in which Windows users can access a PDF file for browser-based reviews:

- Browsing to a file on a network server through My Network Places
- Using the command prompt to access a folder
- Mapping a drive to the network server
- Retrieving the file from a web server

The configuration issues for each of these methods are discussed below.

Browsing My Network Places

Although this method works for small workgroups or networks, it will not necessarily work for large workgroups or networks. The reason is that the file path used in My Network Places may not always be the same for all users. This can happen even though the visual path to the server may appear the same for all users.

For example, suppose you have three users (A, B, and C) in different geographical locations, all commenting on a PDF file located on a file server in Toronto, Canada. If user A is in Hamburg, Germany and browses to the Toronto file server, the IP routing underneath the visual display in Windows will not be the same IP route used by user B in Buenos Aires, Argentina, or user C in Sydney, Australia.

Using the command prompt

Using the command prompt has the same underlying issue as using My Network Places. For example, even though the visual path `\\MyTorontoFileServer\MyPDFFolder\MyPDF.pdf` may appear to be identical to multiple users, each machine may use a different underlying IP route.

Mapping a network drive

The drive letter must be the same for all users. For example, if Machine A is mapped to drive letter E, then all machines commenting on a PDF file located on that server must also be mapped to drive letter E.

The drive letter must also be mapped to exactly the same folder on the server for all of the machines, because the path statement must be identical for all users. For example, if Machine A is mapped to the E: drive and the PDF file is directly located under the E: drive (that is, MyPDFFolder), its path statement would appear as follows:

```
\\MyFileServer\AllPDFFiles\MyPDFFolder\MyPDF.pdf  
E:\MyPDF.pdf
```

However, if Machine B is also mapped to the E: drive, but its mapping is one folder level lower than the PDF file location (that is, AllPDFFiles), its path statement would appear as follows:

```
\\MyFileServer\AllPDFFiles\MyPDFFolder\MyPDF.pdf  
E:\MyPDFFolder\MyPDF.pdf
```

The differing path statements in this example would result in two different hash folders.

Using a web server

A web server is usually the best location for the PDF file. However, in this scheme the network folder is still used to store the comments. This works because the URL or URI is a platform-independent path and does not cause the problems associated with network browsing, using the command prompt, or mapping a network drive.

Since this method uses the file server location for storing comments, the procedures used for mapping a network drive apply.

Network mapping for Mac OS

For browser-based reviews, the path statement must be the same for every Mac OS user that accesses the drive.

For example, suppose Machine A uses the following path:

```
afp//SW2/MyMac/MyPDFFiles/MyPDFFolder/MyPDF.pdf
```

Then Machine B must also use the same path.

On an AppleTalk network, the same machine can appear in more than one AppleTalk Zone, so in this case the path could also appear as follows:

```
afp//SW3/MyMac/MyPDFFiles/MyPDFFolder/MyPDF.pdf
```

In addition, the user can share out multiple levels of folders on the same drive, meaning that the drive itself may be shared out more than once. For example, if the path `SW2/MyMac/MyPDFFiles` exists and has been mounted, it would appear with the volume name `MyPDFFiles`. The users could still browse to the same file using different paths.

As with Windows, the best location for a PDF file that is to be reviewed is a web server, but you would still use the network folder to store comments.

6

Using a SharePoint Repository

For shared reviews, you can use a Windows SharePoint Services document workspace as a comment repository. You set the client to use a SharePoint repository when initiating a review with the shared review wizard.

Reviews using this repository type can be initiated only on a Windows client, but Mac OS clients can participate in the review.

The comment server must be set up with Microsoft Windows Server 2003 and the latest service pack. All reviewers must have access and sufficient rights to the document workspace.

For details on configuring the SharePoint Services server, see its accompanying documentation.

This chapter discusses WebDAV comment repositories and how to configure a client to use one. A WebDAV comment repository can be used for either shared reviews or browser-based reviews.

For browser-based reviews, the WebDAV protocol can be used to send POST and GET commands for adding FDF comment file information to the collaboration server store. WebDAV is the only type of repository that allows Windows users to store their comments on an FTP server.

While configuring clients for a WebDAV repository is simple, the IT configuration can be an issue if the organization's firewall does not support or permit the usage of the WebDAV protocol. Even though a firewall will normally block external access to the WebDAV server, you can configure WebDAV for internal use if you bypass the proxy server by using the exceptions list. For more information, consult your browser documentation. Another option is to use a server that is open to external users.

For more information on setting up a web server to support WebDAV, see <http://www.webdav.org>.

Prerequisites

The following are requirements for the comment repository server:

- The latest version of Apache or Microsoft IIS.
- A site to store the comments. This location must be accessible with read/write rights to all reviewers.
- For browser-based reviews, DAV access must be enabled for the root directory for the comments. For example, to add DAV access to an Apache server with mod_dav installed, add the following code to httpd.conf:

```
<Location /Collab>  
Dav On  
</Location>
```

The following are requirements for each client:

- Windows 2000 or XP, or Mac OS 10 or later.

Manually setting a client to use a WebDAV server

For shared review, you set the client to use a WebDAV server repository when initiating a review with the shared review wizard.

For browser-based reviews, the repository can be set on each client manually as follows.

► **To configure a WebDAV client:**

1. In Acrobat, select **Edit > Preferences > Reviewing**.
2. Change the **Server Type** to WebDAV.

3. In the **Server Settings** field, type the URL/URI using the IP address of the web server, as in the following example: `http://161.43.237.41/WebDAV/`.

Note: The ending back slash is imperative in the URL/URI or this address will not work, because this is the URL/URI naming convention for a WebDAV store. The host name can also be used, but if DNS fails, the host will be unreachable. This applies to both Windows and Mac OS platforms.

For example, suppose John Public and Jane Doe are reviewing a document called `http://www.example.com/docs/document.pdf`, and that the comments will be stored at `http://beta1.example.com868/Collab`.

When they upload their comments to the server, Acrobat creates a hash folder and saves their comments at the following locations:

```
http://beta1.example.com868/Collab/UAtOJErFgHsL2xbLvPPJgC/jpublic.fdf
http://beta1.example.com868/Collab/UAtOJErFgHsL2xbLvPPJgC/jdoe.fdf
```

Automatically setting a client to use a WebDAV server

The comment server must be set up to allow WebDAV access.

For browser-based reviews, an FDF file can be created to configure the comment server settings. The following script sets the default store to `ads`, port `8383` with `/Collab/` as the root, and uses the store type for WebDAV, which is `DAVFDF`:

```
%FDF-1.2
%ããÏÓ
1 0 obj
<<
  /FDF
  <<
    /F (http://reviewserver/pdfs/settingsupdated.pdf)
    /JavaScript
  <<
    After 2 0 R
  >>
>>
endobj
2 0 obj
<<
>>
stream
  Collab.defaultStore = "DAVFDF";
  Collab.setStoreSettings("DAVFDF", "http://ads:8383/Collab/");
endstream
endobj
trailer
<<
  /Root 1 0 R
>>
%%EOF
```

In this sample script, `http://reviewserver/pdfs/settingsupdated.pdf` is the PDF that opens in the browser after the collaboration settings have been set in Acrobat. To adapt this script to your needs, modify the PDF file name and the store settings.

For information on `Collab.defaultStore` and `Collab.setStoreSettings`, see the *JavaScript for Acrobat API Reference* and *Developing Acrobat Applications Using JavaScript*.

Data storage

Acrobat relies upon the underlying mechanism dealing with HTTP connectivity to determine if the server exists. If the DAV server is down or does not exist, the comments cannot be uploaded. Once the users have made the comments and attempt to upload them, Acrobat contacts the DAV server and receives a directory listing of the repository root using the command `PROPFIND`. If the hashed name does not exist, Acrobat will attempt to create one using the command `MKCOL`. If the command fails, Acrobat will return an error message stating that the server is unavailable and that the comments have not been saved.

If the hashed directory already exists or the `MKCOL` command was successful, Acrobat will export the current user's comments to an FDF file. For example, if the current user is logged in as *JPublic*, Acrobat exports only those comments that are authored by *JPublic* to an FDF file named `JPublic.fdf`. This FDF file is then uploaded to the server using the `PUT` command. If the upload is successful, the Send and Receive Comments button will be disabled.

This set of commands only uploads the current user's comments to the server. The user must click either Send and Receive Comments or Receive Comments in order to view comments uploaded by other users. Acrobat does not retrieve the current user's set of comments from the server, since it is assumed that the user is not working at two different machines simultaneously.

Bypassing a proxy server

If your proxy server does not support the WebDAV extension, it is possible to create an exception to the proxy server in order to bypass a firewall. For more information, consult your browser documentation.

8

Using a SOAP Repository

This chapter discusses SOAP repositories and how to configure a client to use one. Acrobat support for the SOAP protocol enables developers to create customized commenting and review workflows using PDF and Acrobat.

Acrobat 6.0 supports SOAP 1.1. Acrobat 7.0 and later supports both SOAP 1.1 and 1.2 by allowing JavaScript clients to perform SOAP request/response transactions via HTTP.

The SOAP protocol supports comment repositories by providing the messaging framework that makes it possible to connect Acrobat with an HTTP-based service such as a web service, thus enabling the exchange of annotation data between Acrobat and the server. Unlike the other commenting workflow methods, the developers setting up the workflow can choose how to store the annotation data in a variety of formats, such as a database, text file, or XML file.

The SOAP support also allows clients to convert Web Services Description Language (WSDL) service descriptions into convenience objects that make SOAP transactions more convenient, and permits SOAP transactions to occur asynchronously. Since SOAP is a standardized XML format, there are a wide variety of server-side tools available.

JavaScript SOAP support

In Acrobat 7.0 and later, JavaScript code can use a WSDL proxy object to generate a SOAP envelope, which can be used to invoke web services on a remote server using either SOAP/RPC or document/literal encoding. There is support for both synchronous and asynchronous connections, and clients can use those connections to perform SOAP request/response transactions via the WSDL proxy object over HTTP or HTTPS.

With the JavaScript `SOAP` object methods, you can locate network services using DNS Service Discovery, bind a service name using a network address and port, connect to the service, and exchange information with the service through either synchronous or asynchronous communication.

For more information on the SOAP protocol, see the W3C documentation at <http://www.w3.org/TR/SOAP>. For more information on WSDL, see <http://www.w3.org/TR/wsdl>.

For more information on the `SOAP` object attributes and methods, see the *JavaScript for Acrobat API Reference* and *Developing Acrobat Applications Using JavaScript*.

Prerequisites

To create a SOAP collaboration server, you need to set up a web server that supports the SOAP 1.1 or 1.2 protocol. Each client must install Acrobat Standard or Professional 6.0 or later, and must have a copy of the collaboration store code.

Manually configuring a client to use a SOAP repository

The Acrobat client can be configured through the use of the JavaScript `SOAP.addAnnotStore` and `SOAP.setStoreSettings` methods. A sample JavaScript file (`sdkSOAPCollabSample.js`) that demonstrates the creation and configuration of a SOAP repository can be found in the Acrobat SDK. The sample provides you with a basis from which you can customize and extend your own SOAP commenting workflows.

The core functions in this JavaScript file are `update`, `enumerate`, and `complete`, as well as a function you would define for reading data from the server and writing data to the server. Descriptions of these are provided in the *JavaScript for Acrobat API Reference* and *Developing Acrobat Applications Using JavaScript*.

As shown in the SDK sample, all annotation data for a PDF document can be composed in one text string and stored in one file on the server. The service should have two Remote Procedure Calls (RPCs): `readFile` and `writeFile`. The `readFile` method returns an XML string containing the annotation data that is automatically deserialized by Acrobat, and the `writeFile` method stores the annotation data in a file stored on the server and returns `true` if successful. Both methods accept a parameter that represents the name of the file on the server containing the annotation data, and the `writeFile` method has a second parameter that represents a string containing an array of annotations, which is automatically serialized by Acrobat. For more information on XML serialization and deserialization, see [“XML serialization and deserialization” on page 34](#).

With the sample JavaScript file installed in the Acrobat JavaScripts folder, a new annotation data store will be created using the `collab.addAnnotStore` method.

► **To configure the SOAP repository:**

1. In Acrobat, select **Edit > Preferences > Reviewing**.
2. Change the **Server Type** to the display name of your SOAP repository, and enter the URL for the SOAP HTTP service in the **Server Settings** text box.

Automatically setting a client to use a SOAP repository

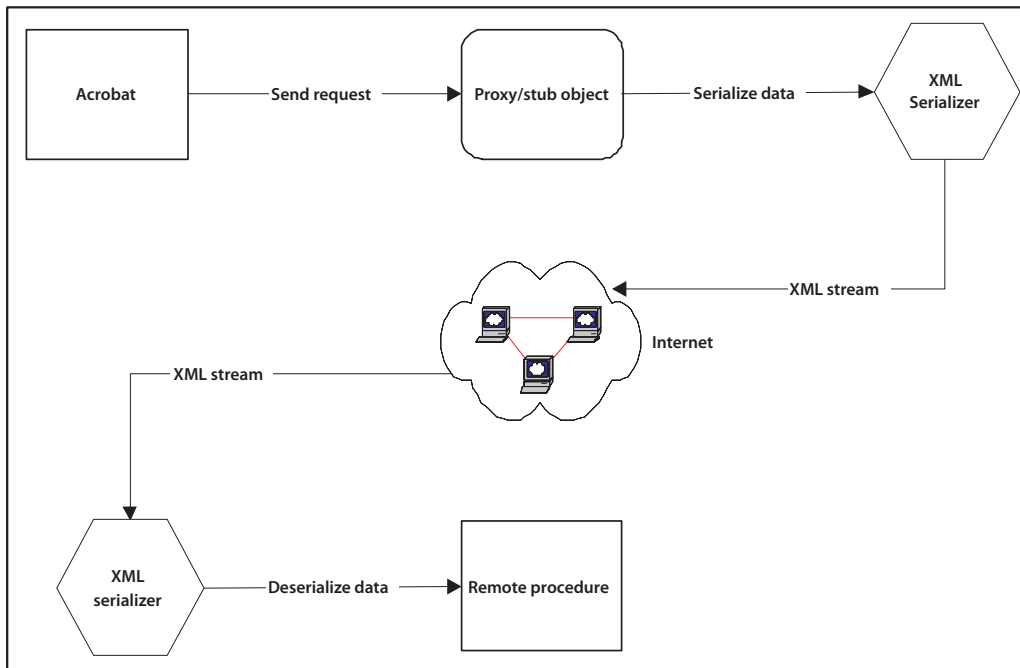
Similar to the approach for a WebDAV repository (see [“Automatically setting a client to use a WebDAV server” on page 30](#)), an FDF file can be created to set the online comment server settings.

Unlike WebDAV, however, Acrobat does not have a predefined store type name for a SOAP repository, since it is a custom solution. It is defined in the JavaScript file stored in the Acrobat JavaScripts folder used to create the SOAP repository, and is used when creating the FDF file containing the JavaScript to configure the store settings.

The store type is the second parameter in the `Collab.addAnnotStore` method. In the sample code given in the `sdkSOAPCollabSample.js` file, where a new SOAP repository is being created, `SDK Soap Sample` is the store type.

XML serialization and deserialization

The SOAP protocol communicates via text-based XML messages. Therefore, before data can be sent across the Internet using SOAP, it must be converted, or serialized, to an XML stream format. Because XML is an open standard, XML streams can be processed by any application, including Acrobat.



When passing parameters to a remote procedure, JavaScript data types are automatically bound to the relevant SOAP data types, and the binding holds true when the remote procedure returns data to the client. This is accomplished through the XML Schema Definition (XSD) language, which provides a standard set of data type definitions that can be exchanged in SOAP envelopes, and is what makes it possible for Acrobat and XML forms to work in an interoperable fashion with web services. For more information, see the *JavaScript for Acrobat API Reference* and *Developing Acrobat Applications Using JavaScript*.

XML serialization is an important consideration in the performance of your workflows, and it is your choice as a developer to decide how the annotation data is stored. For example, you could use a database to store annotation information on the server. This approach would enable you to reduce the amount of XML data exchanged with the SOAP repository by exchanging only subsets of annotation data, known as *lightweight annotations*, with the server. A lightweight annotation contains just enough information to uniquely identify those annotations which must be updated. These contain only the page number, annotation name, author, annotation type, document ID, and modification date.

A *heavyweight annotation* has binary data associated with it. Examples of these include FileAttachment, Sound and Stamp annotations. The sample JavaScript file (`sdkSOAPCollabSample.js`), provided with the Acrobat SDK and demonstrating SOAP connectivity between Acrobat and a SOAP HTTP service, currently supports only lightweight annotations. However, you could customize the code so that binary attachments are exchanged with the web service, either in text format using base64 encoding, or in a more efficient manner using either the SOAP With Attachments (SwA) standard or the Message Transmission Optimization Mechanism (MTOM). This requires that your binary data be represented as a `ReadStream` object. For details, see the *JavaScript for Acrobat API Reference* and *Developing Acrobat Applications Using JavaScript*. It is recommended that the exchange of these heavyweight annotation types be minimized, as encoding large binary files can affect the performance of your application.

9

Designing a Collaboration Interface

Your organization can customize the features in its online collaboration system for browser-based reviews. The options for customization can range from creating a collaboration interface to a large-scale integration of the collaboration system with a document management system. While the specifics of how to fully customize and integrate a collaboration system are beyond the scope of this document, this chapter does suggest some typical customization solutions.

Online collaboration interface design

A simple solution would be to take advantage of the Acrobat online collaboration functionality, which can be used with minimal effort. Solutions based on a network folder repository make it possible to create a collaboration server quickly. The authors are responsible for placing the review document in the proper location and notifying reviewers of that location.

If you prefer to implement a comprehensive collaborative solution, you can build a collaboration interface to facilitate the authors' and reviewers' interactions with the comment repository, thereby streamlining the document review process. The sophistication of the collaboration interface depends on your organization's requirements for its business processes. Since the creation of a collaboration interface involves custom programming and integration, it may be necessary for your organization to contact a solutions provider to assist in the development effort.

A collaboration interface can provide an easily navigable organization of documents, restrictions on document access, and enforcement of the document review workflow. If many documents are to be reviewed, they can be categorized by department or topic area. A collaboration interface can limit access to sensitive design, research, planning, or legal documents, based on an approved list of reviewers. It can also assist authors by notifying reviewers when documents are available for review and automatically sending reminders to tardy reviewers.

Note that document security must be handled before the author submits the document for review.

Questions to consider

Before choosing the type of review and comment interface, there are several important questions to consider:

Will the collaboration interface be organized around documents or around users?

One possible approach is to focus the collaboration interface on document presentation. Users see a list of all documents available for review. These can be organized into different categories, or projects can be created to group the documents appropriately. The collaboration interface can restrict access to sensitive documents through password protection. This approach enables users to comment on a variety of documents related to their expertise or interests.

Another possible approach is to concentrate the presentation of information around users. When users interact with the collaboration interface, they see only the documents that they have been assigned. Thus, a user-centric approach helps reduce the time it takes reviewers to find the documents they wish to review.

How much support should the collaboration interface provide for document authors?

In the collaborative review process, document authors are responsible for transferring their documents to designated locations for review. They would then need to inform reviewers of those locations.

The collaboration interface can be designed to facilitate the author's work. It can automatically transfer the document to the correct location, which can be a network folder or a web server. If a company uses multiple repositories, which would imply that the reviewer's comment repository preferences must be reset, the collaboration interface can provide an FDF file for reviewers to download that automatically sets the repository preferences. A collaboration interface can also notify reviewers of the document's URL and other pertinent information by sending them email from a list provided by the author.

How much support should the collaboration interface provide for the document review and approval workflow?

The document review cycle has a finite duration. In most companies, authors spend time and energy trying to gather feedback from reviewers whose responsiveness may vary. A collaboration interface can track those reviewers whose responses are pending, and send reminder notices to them and their managers. It can also terminate the review cycle by making the document unavailable at the end of the review period.

If the company uses a sequential review process, the collaboration interface can drive the workflow. The solution implementor can design the collaboration interface to prompt the author for an ordered list of reviewers and the amount of time allotted to each to complete the review. The interface can send out notifications and reminders, keep track of who actually commented on the document, and notify the author of the review process status.

Will the organization want formal acknowledgement from reviewers?

Many organizations require that reviewers initial documents after reviewing them. Acrobat supports the use of digital signatures. However, this feature is treated separately from comments, since only comments are stored in the comment repository when a user clicks Send and Receive Comments. Consequently, reviewers must fill in their signature in a separate PDF file, referred to here as the approval PDF document.

An approval PDF document can be handled several ways, depending on the company's document approval workflow.

- Each reviewer could sign an individual approval PDF document and email it to the repository administrator or the author.
- The approval PDF document could be designed with a Submit button that uploads the signed PDF file to the web server. Acrobat has built-in functionality that makes it possible to submit the entire PDF file to a web server. The web server would then save the approval PDF document to an appropriate location.
- The reviewers could download a blank approval PDF document from the web server and insert the approval PDF page or pages into their version of the reviewed document. Once the signed approval pages have been added to the reviewed documents, they can then be sent to the collaboration interface or repository administrator for archival.
- A single approval PDF document can be made available for all reviewers to sign. A checkout system can ensure that only one reviewer at a time actually signs the approval document.

10 Troubleshooting

This chapter provides a number of solutions to common issues when setting up online collaboration.

WebDAV repositories

A warning dialog box states that the collaboration server could not be contacted and that the user is unable to open the site using Internet Explorer.

The server preference settings may be incorrect. The user may have entered the WebDAV URL manually and made an error, as shown in the following example:

```
http://beta1.adobe.abc:8686/Collab
```

when it should be

```
http://beta1.adobe.com:8686/Collab/
```

The trailing slash must be used for it to work properly. Ensure that the server preferences have been specified correctly.

A warning dialog box appears and states that the collaboration server could not be contacted, but the user is able to browse the site.

Acrobat uses special HTTP commands, called DAV verbs, to communicate with the WebDAV server. In some cases, the proxy server may not forward DAV verbs to the server.

If the proxy server does not forward DAV verbs, the network settings must be changed to allow a direct connection to the collaboration server. This can be accomplished in one of three ways:

- Create a PAC script which directs HTTP commands for the collaboration server to bypass the proxy, and then set the user's browser to use the PAC script.
- Edit the user's Internet settings to bypass the proxy for the comments server.
- Edit the user's Internet settings to use a direct connection to the Internet.

All repository types

Acrobat is unresponsive after clicking Send Comments or Go Back Online.

Acrobat cannot connect to the server, possibly because the server is down. This action will eventually time out and Acrobat will display a dialog box stating that the server is unavailable. The server connections should be checked.

Comments only appear intermittently. For example, a user participates in a review session in which the comments appear, and in future sessions observes that those comments are not always present.

This could be an IP translation problem. Sometimes a URL can translate to different IP addresses, depending on the configuration. For example: for the lookup of the address for `www.example.org`, the DNS server will sometimes return 192.239.92.41 and another time may return 208.185.132.75. Since

Acrobat uses the IP address to create the MD5 hash of the URL, a different IP address results in an incorrect MD5 hash.

The solution is to use the IP address when specifying the URL. In this case, for users who want to comment on:

```
http://www.example.org/mypdf.pdf
```

it would be best to use

```
http://192.239.92.41/mypdf.pdf
```

Users do not see each other's comments, yet they are able to upload comments from their own machines successfully.

The problem may be how the users are accessing the PDF on the file server. It is a common practice on Windows to open PDF files in the browser rather than using the File > Open command in Acrobat. The user may surf to the file server using a UNC path such as \\fileserver\users\jsmith\document.pdf, or they may mount a volume such as \\fileserver\users to k: and open the file using k:\jsmith\document.pdf. However, these represent different files in Acrobat because of the different file paths, resulting in different MD5 hashes.

For users to effectively comment on documents that exist on file servers, all users must enter the file name using exactly the same syntax, which is case-sensitive. The best solution to this problem is to use a web page with a link to the PDF.

Comments do not appear to be uploaded from an offline copy.

The user has opened the offline copy in the browser instead of Acrobat. Offline commenting is only possible within Acrobat. The user should upload the comments by clicking Go Back Online.

All server settings are correct and the server is active, yet the user continues to receive the following error message: "Your online comments server is currently unavailable."

If using either Apple Macintosh 9.0 or 9.1, the user name may be longer than the maximum of 31 characters allowed for a file name within those operating system versions. The user should shorten their user name to allow the FDF file name to conform to this limitation.

The Upload Comments button does not activate when double-clicking on an FDF file that was saved using Comments > Export Selected in the web browser.

Make a change to one of the comments and the button will activate.

A message appears saying that the collaboration server is not available.

Try the following solutions:

- Check the client configuration.
- Check the user's network connectivity.
- Check the network configuration. For example, see if a proxy server or firewall is being used.
- Check the client/server configuration.

Users are having a problem with the client configuration.

Check the following:

- Can other machines reach the server via Acrobat Reviewing?
- Are the machines on the same platform?
 - If not, are they using a network folder repository?
 - If not, are they trying to use a Windows-only feature on a Macintosh machine?
- Is the client pointing to the correct type of server?
- Is the client pointing to the correct collaboration store?
- Is the client configured to the host name or the host IP address?
 - If the client is configured to the host name, then try having the customer switch to the host IP address.
- If using WebDAV, has the customer configured the exceptions list for their web browser?
- Does the exceptions list use both the host name and the host IP address?

Network problems

Users are having problems with the network.

Check the following:

- Is the user connected to the network?
- If the client is configured to the host name address, then check to see if the user can 'ping' the host name.
- If the client is configured to the host IP address, then check to see if the user can 'ping' the host IP address.
- If the user cannot ping either host name or the host IP address, then it is a networking issue and the user's IT department should be contacted.
- If you would like to continue helping the user, you can run `tracert` to the host name or host IP address of the server to determine where the connection fails.
- Is the user able to access the collaboration store via the Windows command prompt, My Network Places, or the Chooser?
 - If not, it may be a connectivity issue, and the user's IT department should be contacted.
- Does the user's company use a proxy server?
- Does the proxy server support WebDAV?
- Have they configured the exceptions list for their web browser?
- Does the exceptions list use both the host name and the host IP address?
- Is the user trying to access a server outside of their firewall?
 - If so, then it is a networking issue for the user's IT department.

Client/server configuration and permissions

The user is having a problem with the client/server configuration and permissions.

Check the following:

- Is the user able to access the collaboration store via the Windows command prompt, My Network Places, the Mac OS X Go menu, or the Mac OS Chooser from that machine?
 - If not, it may be a permissions issue, and the user's IT department should be contacted.
- Can the user access the collaboration store from other machines via the Windows command prompt, My Network Places, the Mac OS X Go menu, or the Mac OS Chooser?
 - If not, it may be a permissions issue, and the user's IT department should be contacted.
 - If so, it may be a 'Trust' issue.
- When using the database comments feature, does the user have a valid account with read, write and execute permissions on the database and table?
 - If not, or if the user does not know, the user's IT department should be contacted.
- The FDF file is being downloaded but will not execute from Mac OS. Is the server's MIME configuration correctly set for the FDF file type?
- The user cannot access the network folder. Does the user have proper read, write and execute permissions on the server share?
- The user cannot access the WebDAV store on the web server. Does the user have proper read, write and execute permissions on the server store?
- Does the network use a proxy server or firewall? Does it allow the WebDAV extensions to the HTTP 1.1 protocol?
- The user cannot access the server share. Is the server internal or external to the network firewall?

Index

- A**
 - Acrobat Tracker 22
 - advertising a service 19
- B**
 - browser-based reviews
 - description 6
 - preference configuration 15
- C**
 - client compatibility 7
 - client configuration
 - FDf files 15
 - InstallShield Tuner 15
 - JavaScript 16
 - manual 16
 - network folder repositories 23, 24
 - SOAP repositories 33
 - WebDAV repositories 29
 - collaboration interface design 35
 - comment repositories
 - choosing 9
 - comparing 9
 - comparison table 12
 - overview 7
 - requirements 9
 - resources 9
 - types 7
 - comments in network folder and WebDAV repositories 13
 - compatibility of clients 7
- D**
 - data structures 18
 - designing interfaces 35
 - DNS 19
- E**
 - Enterprise Service Discovery 19
 - extensibility 7
- F**
 - FDf files 8, 13, 15, 30
 - firewalls 29
- H**
 - hash algorithm 13
- I**
 - InstallShield Tuner 15
 - interface design 35
- J**
 - JavaScript 8, 16, 32
- M**
 - managing reviews 22
 - mapping 25, 27
 - MD5 hash algorithm 13
- N**
 - network folder repositories
 - automatic configuration 24
 - choosing 9
 - client configuration 23
 - handling comments 13
 - path issues 25
 - prerequisites 23
 - network mapping 25, 27
 - network path issues 25
- P**
 - path issues 25
 - PDF open parameters 18
 - proxy servers 29, 31
- R**
 - repositories, See comment repositories
 - RSS 22
- S**
 - security 17
 - service discovery 19
 - shared reviews 6
 - SharePoint Services repositories 11, 28
 - SOAP repositories
 - automatic configuration 33
 - choosing 11
 - client configuration 33
 - JavaScript support 32
 - prerequisites 32
 - XML 34

T

Tracker 22
troubleshooting 37

U

user identity 17

W

WebDAV repositories
 automatic configuration 30
 choosing 10
 client configuration 29
 data storage 31
 handling comments 13
 prerequisites 29
 proxy servers 31
WSDL 32

X

XML 11, 34