TECHNICAL PAPER

Adobe® Flash® Player 10 Administration Guide

Version 1.0
October 2008

RavenWhite EX2018
Home Depot v. RavenWhite
IPR2024-01316

© 2008 Adobe Systems Incorporated. All rights reserved.

Adobe® Flash® Player 10 Administration Guide

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.

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

Adobe, the Adobe logo, ActionScript, and Flash 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 Inc., registered in the United States and other countries. Windows is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

MPEG Layer-3 audio compression technology licensed by Fraunhofer IIS and Thomson Multimedia (http://www.mp3licensing.com)

Speech compression and decompression technology licensed from Nellymoser, Inc. (www.nellymoser.com).

Video compression and decompression is powered by On2 TrueMotion video technology. © 1992-2005 On2 Technologies, Inc. All Rights Reserved. http://www.on2.com.

This product includes software developed by the OpenSymphony Group (http://www.opensymphony.com/).

This product contains either BSAFE and/or TIPEM software by RSA Security, Inc.

Sorenson Spark™ video compression and decompression technology licensed from Sorenson Media, Inc.

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

Introduction 5

Why install Flash Player? 5
Intended audience 6
Documentation map 6
Additional resources 6
Flash Player and deployment 7
Design and development tools 7

Chapter 1: Flash Player Environment 9

Player files and locations 9
Firefox/Mozilla plug-in architecture 10
ActiveX Control on Windows 10
Additional files 11
Data formats used 12
Network protocols used 13
Player processes 13
Player versions 14

Chapter 2: Player Installation 17

Uninstalling Flash Player 17
Uninstalling on Windows 18
Uninstalling on Linux 18
Uninstalling on Macintosh 18
EXE installation 18
Active Directory installation 19
Configuring SMS 20
SMS and Adobe Catalog installation 27
System requirements for SMS deployment 28
SMS tools for deploying custom updates 28
Downloading the Flash Player catalog 28
Importing the Flash Player catalog 28
Publishing the Flash Player catalog 29
Confirming successful publication 30
Deploying the update 30
Additional resources 32

Interactive MSI installation using SMS 32
Command line MSI installations 53
Windows registry keys 56
DMG installation for Macintosh 56
Customizing player behavior 56
Troubleshooting installation problems 57
Additional resources 57

Chapter 3: Administrator Settings 59

Privacy and security settings (mms.cfg) 59
What's new 59
mms.cfg file location 60
Setting options in the mms.cfg file 60
Privacy options 63
User interface option 64
Data loading and storage options 64
Update options 67
Security options 69
Socket connection options 72
GPU Compositing 72
RTMFP options 73
The Global FlashPlayerTrust directory 74

Chapter 4: User-Configured Settings 77

Accessing user settings 77
Privacy options 78
Local storage options 79
Update options 80
Security options 80
The User FlashPlayerTrust directory 82

Chapter 5: Security Considerations 85

Security overview 85
Security sandboxes for local content 87
The local-with-file-system sandbox 88
The local-with-networking sandbox 88
The local-trusted sandbox 89
About compatibility with previous Flash Player security models 89
Data loading through different domains 90
Additional security resources 91

Introduction

Welcome to the Adobe® Flash® Player Administration Guide for Flash Player 10. This document describes Flash Player, how it's installed, how it works, and how it can be controlled to suit the needs of a specific network environment. Read this document if you are an IT or administrative professional who manages the installation or use of Flash Player for multiple users in a controlled environment.

For the latest version of this guide, see the Adobe Flash Player Administration Guide section of the Flash Player Developer Center at www.adobe.com/go/flash_player_admin.

To deploy the player, you must first acquire a license to do so. Distribution licenses are free of charge and can be acquired through the online licensing application at www.adobe.com/licensing/distribution. For answers to questions regarding Flash Player licensing and deployment, see the Adobe Player Distribution FAQ at www.adobe.com/licensing/distribution/faq.

This chapter contains the following sections:

Why install Flash Player? 5
Intended audience 6
Documentation map 6
Additional resources 6

Why install Flash Player?

Flash Player is the software that allows computers to play multimedia content contained in SWF (pronounced "swiff") files, which are the main type of file used by Flash Player. This content can be created by Adobe Flash Professional, Adobe Flex Builder, or other tools that output the SWF file format. SWF content can range from simple animations to online advertisements to complete applications that communicate over the Internet. Flash Player is available in multiple forms. In its most popular form, it is embedded in a web browser as a plug-in or an ActiveX control.

You may have been asked to deploy Flash Player in your network environment because someone in your company has built a SWF application for business use, or because there is external SWF content that employees want to have access to.

Intended audience

This document is intended for the following audience:

- IT administrators who need to deploy Flash Player on their network computers.
- Developers (including programmers and other authors) designing and publishing SWF applications who want to understand the implications of SWF content deployment in their network environment.
- IT managers interested in the security of SWF applications in their network environment.

This document assumes that the reader is familiar with Flash Player and with Adobe® ActionScript®, along with related terms, authoring tools, and environments.

Documentation map

This document provides information about the following topics:

- Where files are placed during Flash Player installation process, where SWF applications store data locally, and how to determine which version of the player is installed (see Chapter 1, "Flash Player Environment," on page 9)
- The installation process (see Chapter 2, "Player Installation," on page 17)
- Customizing player settings (see Chapter 3, "Administrator Settings," on page 57)
- Settings that the user can specify (see Chapter 4, "User-Configured Settings," on page 73)
- Security considerations (see Chapter 5, "Security Considerations," on page 81)

Additional resources

The following sites provide information about some general topics related to the Flash Platform, Flash Player, and design and development tools. For information about sites related specifically to issues covered in this document, see the chapter that covers that issue. For example, for an extensive list of resources specific to the topic of security, see "Additional security resources" in Chapter 5, "Security Considerations."

Flash Player and deployment

The following sites contain information and links to help you understand how to deploy Flash Player and work with SWF files.

- The Flash Player Support Center at www.adobe.com/support/flashplayer/ provides information on a number of topics relating to installing, using, and deploying Flash Player. It also contains links to documents that can answer just about any question you might have about Flash Player, locations for downloading the player, user forums, and so on. Much of the information in this document is excerpted from documents available from the Support Center.
- The Flash Player Developer Center at www.adobe.com/devnet/flashplayer provides extensive information about Flash Player, including development and deployment of applications. The content includes Tech Notes, articles, and tutorials.
- The SWF File Format Specification at www.adobe.com/go/swf_file_format documents the SWF file format and describes how to write SWF files.
- The Flash Player Release notes at www.adobe.com/support/documentation/en/flashplayer/releasenotes.html contain information about features, fixes and improvements, and known issues for each version of the player.

Design and development tools

Adobe provides the following tools for developing SWF files (the file format that executes in Flash Player):

- Adobe Flash CS4 Professional (www.adobe.com/products/flash/)

  In Flash CS4 Professional, designers and developers create FLA files that contain graphical elements, a timeline, and ActionScript code. Both ActionScript 2.0 and ActionScript 3.0 are supported. FLA files are compiled into SWF files.
- Adobe® Flex® (www.adobe.com/products/flex/)

  In Flex, developers create MXML files that describe the visual and code elements of their applications. They can also use ActionScript 3.0. Both MXML and ActionScript compile into SWF files.

CHAPTER 1

Flash Player Environment

1

This chapter describes the different environments in which Adobe Flash Player runs, where Flash Player files are stored on the system, processes Flash Player generates, and information on determining which version of the player is installed on a system.

This chapter contains the following sections:

Player files and locations 9
Data formats used 12
Network protocols used 13
Player processes 13
Player versions 14

Player files and locations

Adobe Flash Player is normally deployed as a browser plug-in or ActiveX control. For each player environment, two versions of Flash Player are available—a "Content Debugger" version for developers, and a "Release" version for end users. The Content Debugger player is installed with the development environment. This player implements the same feature set as the Release player, but also displays run-time errors during compilation. Each of these implementations is described in this section.

NOTE	There is also a stand-alone player, but it's usually installed by the development tools, not deployed by administrators.

Firefox/Mozilla plug-in architecture

Mozilla, Mozilla-based browsers (such as Firefox), and the Safari browser on the Macintosh use this plug-in.

Windows plug-in filenames and locations

On Windows, files named NPSWF32.dll and flashplayer.xpt are installed. These files are placed in the following directory, along with the ActiveX control. For example:

%WINDIR%\System32\Macromed\Flash

NOTE	The %WINDIR% location represents the Windows system directory, such as C:\WINDOWS.

The Windows plug-in installer also places a broker application called NPSWF32_FlashUtil.exe in the same directory as the Flash Player Plug-in DLL. NPSWF32_FlashUtil.exe includes functionality required by Windows Vista and by the auto-update notification process.

Macintosh plug-in filenames and locations

On the Macintosh, files named Flash Player.plugin and flashplayer.xpt are installed. These files are placed in the Internet plug-ins folder in the Library folder.

Linux plug-in filenames and locations

On Linux, files named libflashplayer.so and flashplayer.xpt are installed. These files are placed in the /usr/lib/flash-plugin/ directory, and links to them in /usr/lib/.

ActiveX Control on Windows

The ActiveX control is used by Microsoft Internet Explorer as well as certain other applications, such as Microsoft Powerpoint and Yahoo Messenger. The player is an OCX file whose name reflects the version number and unique letter for each subsequent release (if any) of the player. For example, for the initial release of Flash Player 10, the Release player filename is Flash10a.ocx and the Content Debugger player filename is FlDbg10a.ocx. In later releases, these filenames might change to Flash10b.ocx and FlDbg10b.ocx, then to Flash10c.ocx and FlDbg10c.ocx, and so on.

The OCX files are stored in the following directory:

%WINDIR%\System32\Macromed\Flash

NOTE	The %WINDIR% location represents the Windows system directory, such as C:\WINDOWS.

Additional files

When Flash Player is installed on Windows, certain utility files are installed that perform special functions for Flash Player, including auto-update notification and brokering certain processes on Windows Vista. These utility files also provide developers a way to easily switch between player versions during testing. This functionality is briefly described in this section, and more information is available in the TechNote entitled "Installation issues when switching between release and debugger versions during development and testing" at www.adobe.com/go/4da116d3.

FlashUtil.exe and GetFlash.exe

A utility file named FlashUtil*nn*.exe is installed with Flash Player in the %WINDIR%\System32\Macromed\Flash directory. The utility is versioned with the control; for example, FlashUtil10.exe is installed with the control Flash10.ocx. The FlashUtil*nn*.exe file is associated with the auto-update functionality.

When the browser plug-in is installed, a similar application named NPSWF32_FlashUtil.exe is installed. This file is associated only with the browser plug-in, and is separate from the FlashUtil.exe used for the ActiveX control.

Data formats used

Several file types are created or read by Flash Player. These file types are summarized in the following list.

- SWF - The SWF file format (pronounced "swiff") is an efficient delivery format that contains vector graphics, text, video, and sound. Flash Player executes SWF files. SWF files can be loaded into Flash Player dynamically by instructions in other SWF files.
- CFG - These are configuration files that network administrators and developers can deploy along with Flash Player to customize Flash Player settings and address certain security issues for all users. For more information, see Chapter 3, "Administrator Settings," on page 59. End users can also create CFG files to address certain security issues for that specific user; see "The User FlashPlayerTrust directory" on page 82.
- SWC (pronounced "swik") - These are SWF files that developers deliver as components for use when working in the Flash authoring environment.
- SO - Shared object files are used by Flash Player to store data locally. For example, a developer may create a game application that stores information on high scores. This data may be stored either for the duration of a Flash Player session, or persistently across sessions. In addition, Flash Player creates a persistent shared object that stores player settings, such as the amount of disk space a web site can use, if any, when creating shared objects. Shared object files are stored in the following location:
  - **Windows Vista** C:\Users*username*\Application Data\Macromedia\Flash Player\#SharedObjects*randomDirectoryName*
  - **Windows 2000 and Windows XP** C:\Documents and Settings*username*\Application Data\Macromedia\Flash Player\#SharedObjects*randomDirectoryName*
  - **Macintosh** /Users/*username*/Library/Preferences/Macromedia/Flash Player/#SharedObjects/*randomDirectoryName*
  - **Linux** GNU-Linux ~/.macromedia#SharedObjects/*randomDirectoryName*

  Shared objects are stored in a directory with a randomly generated name for security purposes. Flash Player remembers how to direct a SWF file to the appropriate location, but users of other applications outside Flash Player, such as a web browser, cannot use those applications to access the data. This limitation ensures that the data is used only for its intended purpose.
- MP3 - The compressed audio file format.
- JPG, PNG, and GIF- Image file formats. The TIF and BMP formats are not directly supported for use in SWF files.
- FLV - Flash Player compressed video format.

- XML (eXtensible Markup Language) - Used for sending and receiving larger amounts of data with structured text.
- MXML - The XML-based language that developers use to lay out components in Flex applications.

NOTE	If you block access to any of these file types, certain functionality of Flash Player may be disabled.

Network protocols used

Flash Player can use the following network protocols:

- HTTP
- HTTPS
- RTMP (Real Time Messaging Protocol) - a proprietary protocol used with Flash Media Server to stream audio and video over the web. The default connection port is 1935.
- RTMPT - RTMP tunneling via HTTP. The default connection port is 80.
- RTMPS - RTMP tunneling via HTTPS. The default connection port is 443. (For more information about using the RTMP protocols, see the TechNote entitled "HTTP Tunneling Protocols" at www.adobe.com/go/tn_16631.)
- SOAP - Simple Object Access Protocol
- UNC - Universal Naming Convention, such as file:///C|/Flash Files/filename.swf.
- TCP/IP - Transmission Control Protocol/Internet Protocol
- FTP - File Transfer Protocol
- SMB - Server Message Block. SMB is a message format used by DOS and Windows to share files, directories, and devices. Flash Player can load animations and SWF files from remote SMB shares. Flash has restrictions on what Flash SWF files loaded from SMB shares are allowed to do.
- SSL - Secure Sockets Layer
- AMF - ActionScript Message Format

Player processes

Most often, Flash Player runs as a browser plug-in. When Flash Player operates in this mode, it does not launch any new processes on the end user's computer. When run as a stand-alone player, it launches a process named FlashPlayer.exe.

Flash and Flex developers can package their SWF files into stand-alone EXE files, called projectors. When a projector is run, it launches a single process, named for the projector executable filename.

Other processes are created when Flash Player auto update occurs. GetFlash.exe or FlashUtil.exe will be running during an auto update request and subsequent downloading and installing of the updated player.

Player versions

Before deploying the player, you might want to know what version is already installed on an end user's machine. An easy way to determine the version of Flash Player installed is to navigate to www.adobe.com/products/flash/about; this page displays a message stating which version is installed. Or, while a SWF file is playing, right-click (Windows or Linux) or Command-click (Macintosh) on the SWF content and then choose "About Flash Player" from the context menu.

On the Macintosh, you can navigate to the Flash Player.plugin file located in the /Library/Internet plug-ins folder, then Command-click and choose Get Info. The version number is available on the General menu.

On Windows, you can determine which version of the ActiveX control is installed by navigating to the directory where the OCX file is located (see "ActiveX Control on Windows" on page 10 for the default location). Right-click on the OCX file and choose Properties, then inspect the value in the Version tab. If the OCX file isn't installed in the default location, you can determine its location and name by inspecting the following registry key, which is created when the OCX control is registered:

```
HKEY_CLASSES_ROOT\CLSID\{D27CDB6E-AE6D-11cf-96B8-
  444553540000}\InprocServer32
```

Similarly, you can determine the Plug-in version by examining the version tab of the NPSWF32.dll file, which is located in the same folder as the ActiveX control.

End users can view the currently installed version by visiting www.adobe.com/products/flash/about with their browser, or by right-clicking on a SWF file and selecting "About Flash Player..."

For information on how to incorporate player version detection into web sites, see the "Detection and Installation" section at the Flash Player Developer Center (www.adobe.com/devnet/flashplayer/detection_installation.html).

If you want to learn which version of Flash Player is installed on an end user's machine without going to each machine individually, you or a developer at your site can create and distribute a SWF file that implements that `System.Capabilities.version` API and reports the results to a database using a command such as `HTTP GET` or `POST`. This technique is useful for activities such as collecting statistics on how many users have which version of Flash Player.

CHAPTER 2

Player Installation

2

The licensed installers for Flash Player are available in a number of forms. For Windows Internet Explorer (ActiveX) and Firefox/Mozilla plug-ins, you can download an executable installer (EXE file) or an MSI installer.

If you are using Microsoft Systems Management Server (SMS) 2003 R2, you can also import the Adobe Flash Player Catalog with the Inventory Tool for Custom Updates.

For Macintosh OS X, you use a DMG installer. For Linux, you use an RPM installer.

Adobe strongly recommends that you implement network installation strategies in a testing environment prior to implementation in a live environment. Adobe support cannot provide troubleshooting assistance for customized installations.

This chapter includes the following sections:

Uninstalling Flash Player 17
EXE installation 18
Active Directory installation 19
Configuring SMS 20
SMS and Adobe Catalog installation 27
Interactive MSI installation using SMS 32
Command line MSI installations 53
Windows registry keys 56
DMG installation for Macintosh 56
Customizing player behavior 56
Troubleshooting installation problems 57
Additional resources 57

Uninstalling Flash Player

Before a new version of Flash Player is installed, you might want to uninstall any existing Flash players.

Uninstalling on Windows

Before uninstalling Flash Player, be certain to quit **all** running applications, including all Internet Explorer windows, AOL Instant Messenger, Yahoo Messenger, MSN Messenger or other Messengers. Check the Windows system tray carefully to make certain no applications that might possibly use Flash Player are still in memory.

Use the uninstaller available at www.adobe.com/go/tn_14157 to uninstall any version of the player. If you want to uninstall in silent mode, use the "/silent" or the "/s" command-line parameter:

```
uninstall_flash_player.exe /silent
```

Uninstalling on Linux

To uninstall Flash Player on Linux, log in as root and use the following command:

```
rpm -e flash-plugin
```

Uninstalling on Macintosh

To uninstall Flash Player on the Macintosh, make sure all browsers are closed, along with any programs that might be running SWF content, such as the Dashboard. Then remove the Flash Player.plugin and Flash Player Enabler.plugin (if it exists) from the /Library/Internet Plug-ins folder. If you prefer, you can download uninstallers that are available at www.adobe.com/go/tn_14157.

EXE installation

The EXE installer can be run in either of two modes, interactive or silent. The interactive mode presents a full user interface and displays error dialogs if necessary. The silent mode does not present a user interface, and returns error codes if necessary.

To run the EXE in silent mode, use the "/silent" or the "/s" command line parameter:

```
path to installer\install_flash_player_active_x.exe /s
```

The following error codes are returned if the installation fails:

- 3 - Does not have admin permissions
- 4 - Unsupported OS
- 5 - Previously installed with elevated permissions
- 6 - Insufficient Disk Space
- 7 - Trying to install older revision
- 8 - Browser is open

Active Directory installation

To deploy the Flash Player MSI through the Active Directory, you use group policies. Also, the MSI for Flash Player must exist within a network share on which everyone has read permissions.

Flash Player can be deployed to either computers or users.

- You can publish Flash Player to users.

  Publishing is a group policy action.Therefore, when you publish Flash Player it doesn't install the MSI, but it does make it available to users the next time they log in. This implementation gives the user the choice to install Flash Player through the Add/Remove Programs option in the Control Panel.
- You can assign Flash Player to users.

  Assigning Flash Player to users is like publishing in that it is also a group policy action; the assignment does not take effect until the next time that the user logs in. However, unlike publishing, when the user logs in, Flash Player will be installed and an icon added to the desktop.
- You can assign Flash Player to computers.

  Assigning Flash Player to a computer works similarly to assigning it to a user, with two major differences. First, the assignment is linked to the computer and not to the user; it takes effect the next time that the computer is restarted. The second difference is that the deployment process actually installs Flash Player.

To perform the deployment, open the Group Policy Editor.

To publish or assign an application to a user:

1. Navigate through the group policy console.
2. Select User Configuration > Software Settings > Software Installation.
3. Right-click on the Software Installation container

4. Select the New > Package commands from the context menu.
5. Select the Flash Player MSI and select Open.
6. Choose if you want to publish or assign Flash Player.
7. Select OK.

To assign Flash Player to a computer:

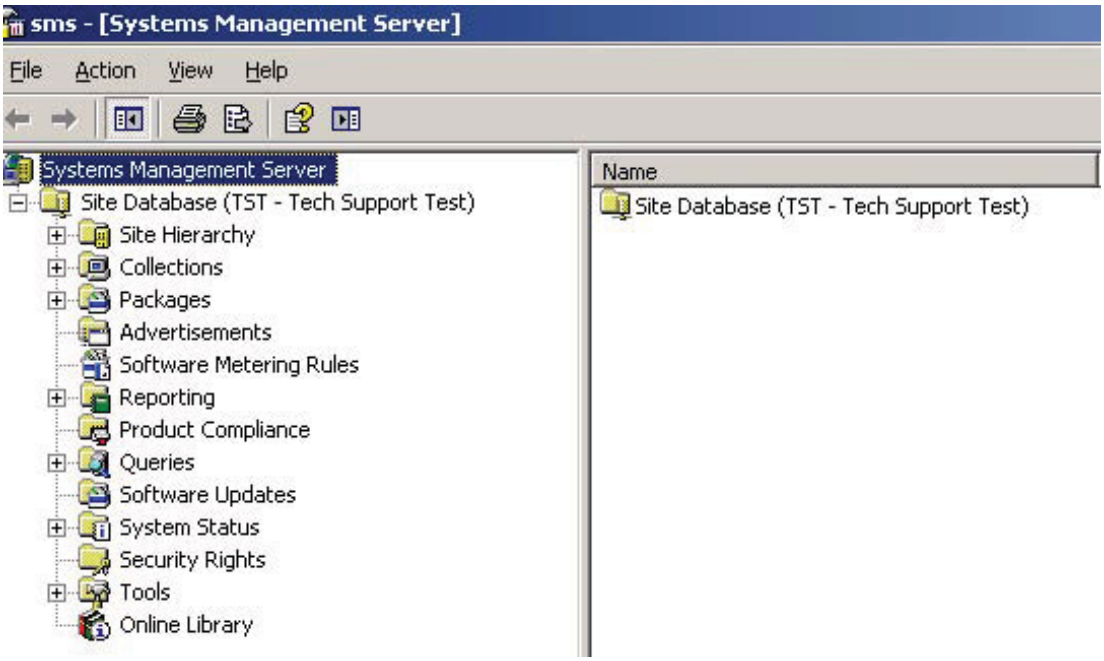
1. Navigate through the group policy console.
2. Select Computer Configuration > Software Settings > Software Installation.
3. Right-click on the Software Installation container.
4. Select the New > Package commands from the context menu.
5. Select the Flash Player MSI and select Open.
6. Choose to assign Flash Player.
7. Select OK.

You can see that the instructions to assign Flash Player to a user or to a computer are similar. The main difference is selecting the user or computer configuration in step two.

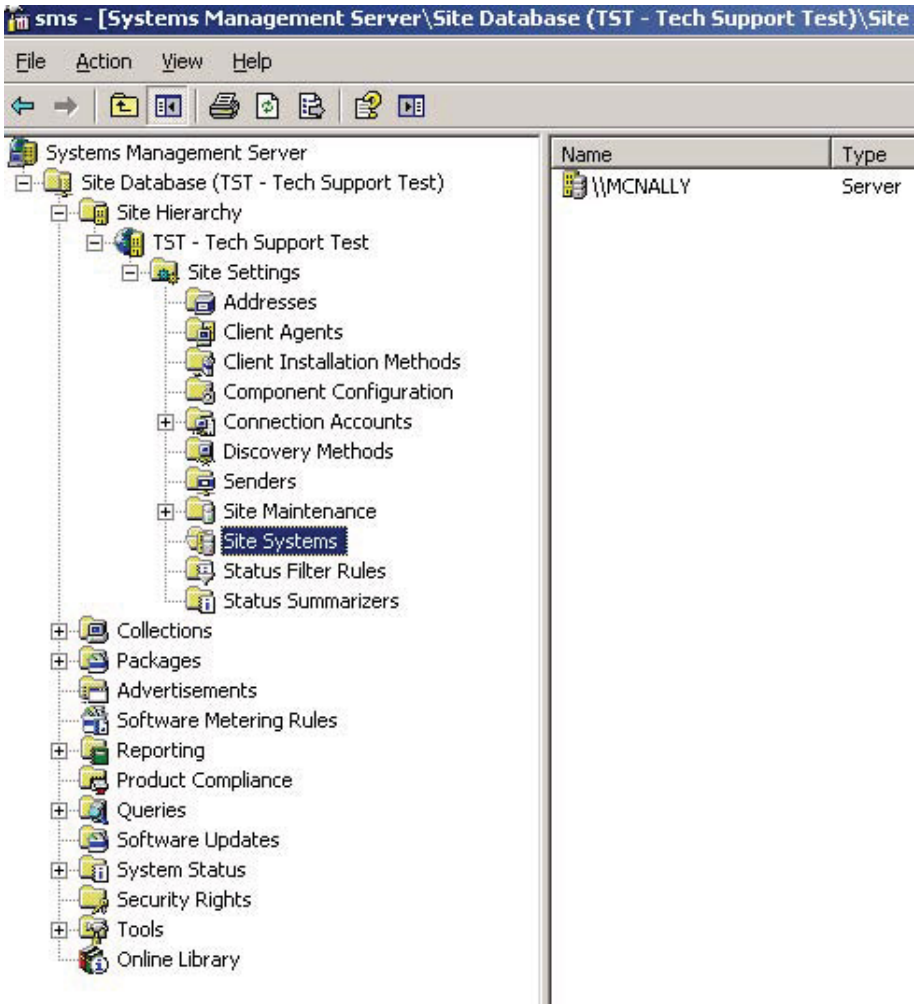
Configuring SMS

If you plan to use SMS to deploy the player, using either the Adobe Catalog or the MSI file, follow these instructions before starting the deployment process.

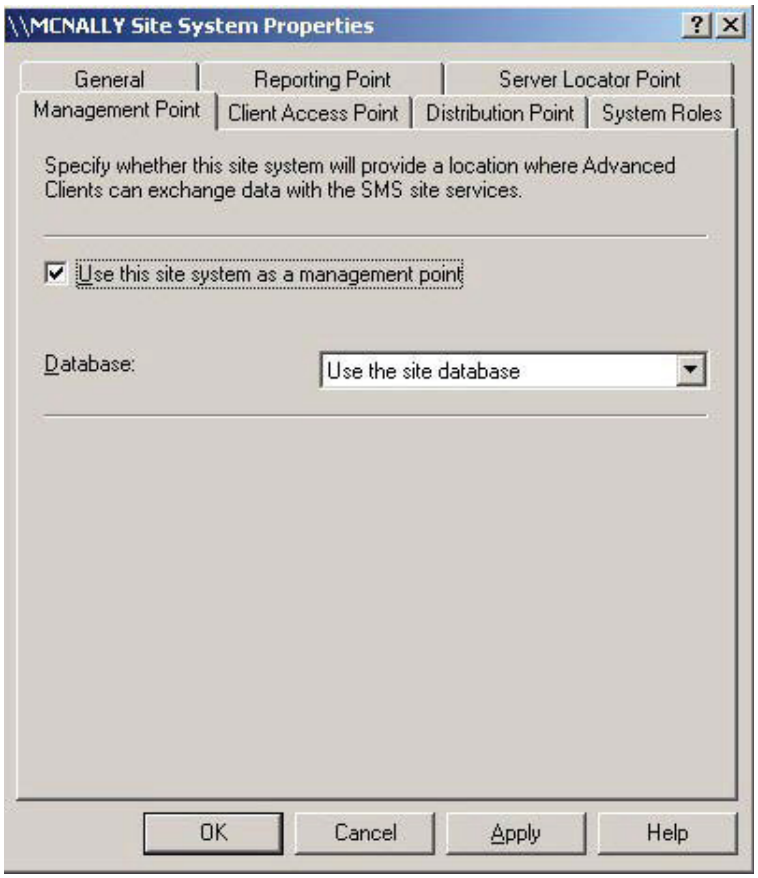
1. Start the SMS Administrator Console.



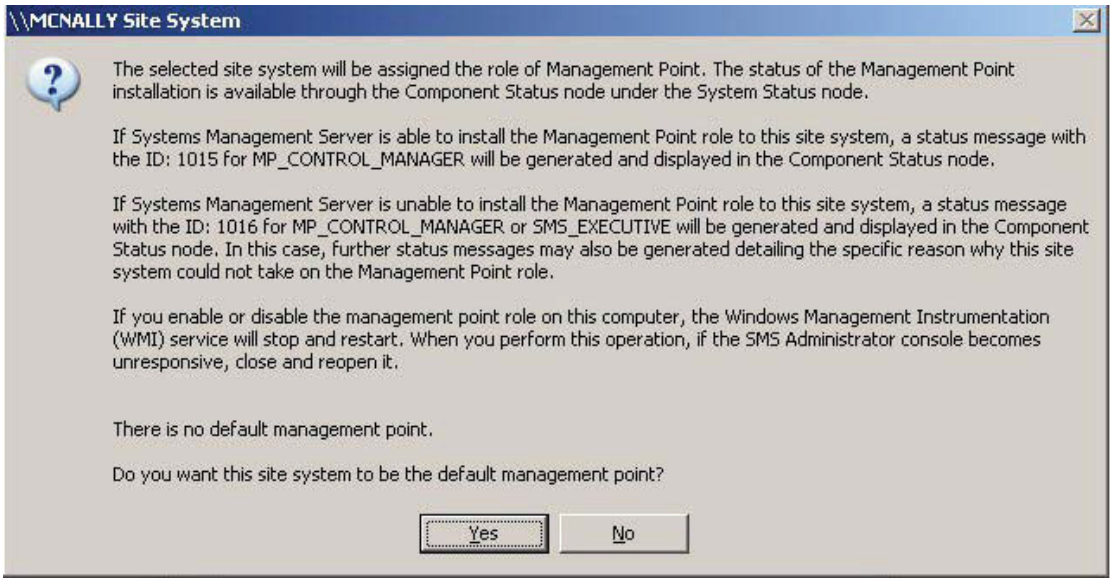
2. Expand the Site Hierarchy, select Site System, and double-click on the SMS site server. (In this example the site server is \\MCNALLY)



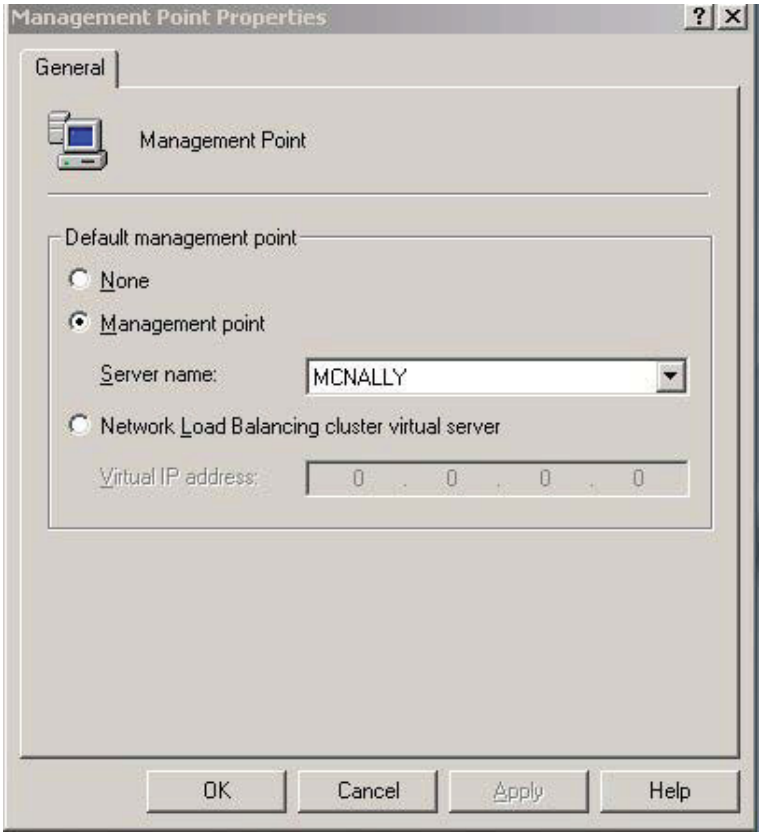
3. Confirm that “Use this site system as a management point” is enabled.



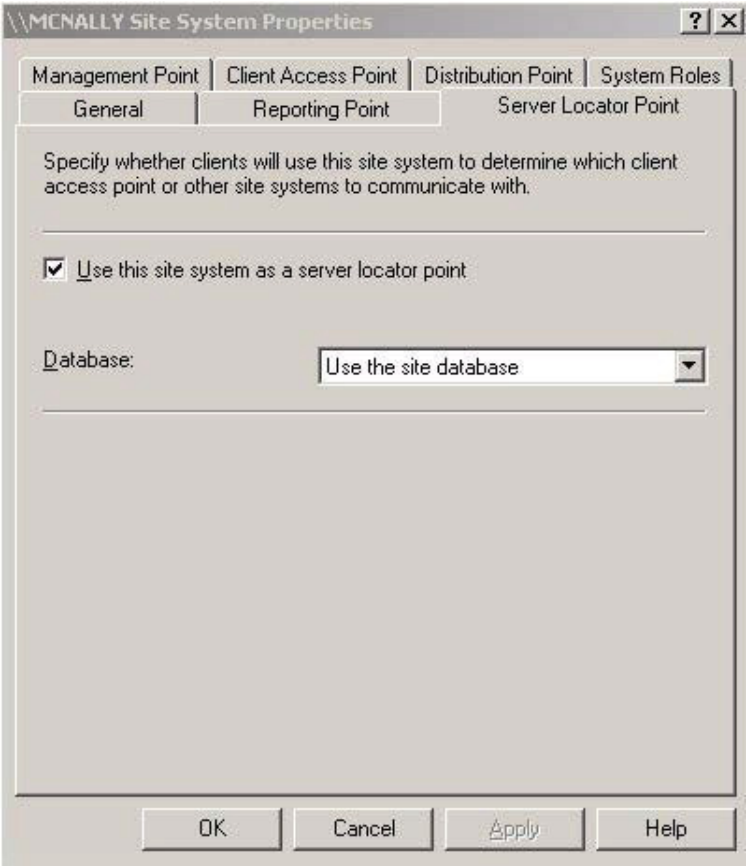
4. If you have not yet selected the default management point, the following error message is displayed.



Select Yes to continue, then select Component Configuration, and then select Management Point. This server is now set to be the default Management Point for your site.

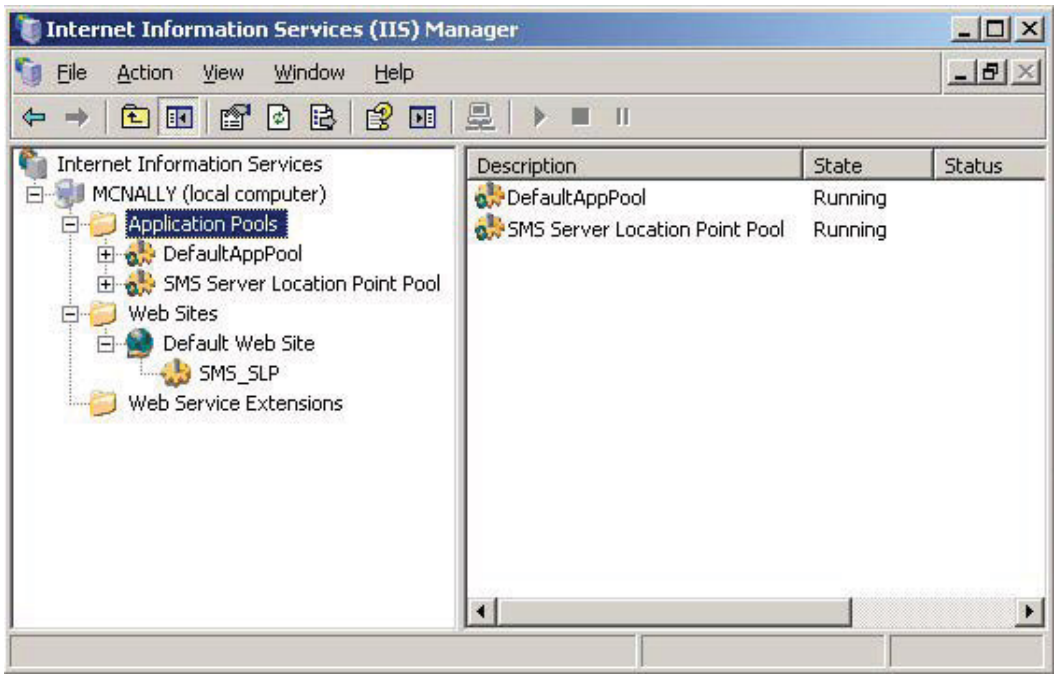

Management Point Properties
General
Management Point
Default management point
None
Management point
Server name:
MCNALLY
Network Load Balancing cluster virtual server
Virtual IP address:
0 . 0 . 0 . 0
OK
Cancel
Apply
Help

5. If necessary, reopen the Site System Properties. Then, on the Server Locator Point tab, enable “Use this site system as a server locator point”.



This setting helps the client find the site server.

6. Select Start, All Programs, Administrative Tools, Internet Information Services (IIS) Manager.



Notice that your website that was added to the IIS Manager.

7. As a final step, you may also want to set up some Discovery Methods in the SMS Administrative Console, so your site will generate collections (machines or user ID's) automatically.

SMS and Adobe Catalog installation

SMS 2003 R2 includes two new powerful tools for software deployment—the Inventory Tool for Custom Updates (ITCU) and the Custom Updates Publishing Tool (CUPT). This section briefly describes these tools and explains how to use them to deploy Flash Player.

NOTE	Installation using SMS can fail if the player is being installed on a machine where the logged-in user does not have administrative privileges. For information on resolving this issue, see the TechNote entitled "Flash Player MSI installation will fail on machines that don't have administrative privileges" at www.adobe.com/go/df875c9e.

System requirements for SMS deployment

To use SMS 2003 R2, the hierarchy, including clients, must be updated to SMS 2003 Service Pack 2 (SP2). In addition, to use the CUPT, you must be running the Microsoft Management Console (MMC) 3.0 or higher. You do not have to install CUPT on the SMS Site Server, but it must be installed on at least one Windows XP machine. The CUPT requires SQL Server 2005 for hosting its database. If SQL Server 2005 is not available, SQL Server Express Edition can be used. The CUPT tool allows administrators to managing custom updates in the SMS system and it also has features to test created catalogs before publishing them in SMS.

SMS tools for deploying custom updates

The ITCU is a new inventory tool that works with custom update catalogs such as the Adobe catalog. ITCU creates custom collections, packages, and advertisements that are used for deploying the scan tools to SMS clients in the enterprise. ITCU retrieves the catalog, in this case the custom updates catalog, from an accessible SMS distribution point, perform the scan based on catalog data, insert the results of that scan into Windows Management Instrumentation (WMI), and report the results via hardware inventory.

Custom updates using the CUPT can take two forms—updates that are provided by third-party vendors for software they produce, such as Adobe, and updates created internally that are unique to a particular environment. These updates are distributed as *catalogs*. Using third-party updates is a simple matter of downloading the catalogs and adding them to SMS.

Downloading the Flash Player catalog

Adobe provides a Flash Player Catalog named AdobeFlashPlayerCatalog.cab for licensing and use with SMS 2003 R2. You can download the catalog from your licensed download page. After you download the catalog, you import it into the CUPT and then publish it to SMS. The rest of this section explains how to perform these tasks.

Importing the Flash Player catalog

Follow these steps to import the Flash Player catalog into SMS.

1. Select Start, All Programs and choose Systems Management Server.
2. Select Custom Updates, then choose Publishing Tool to launch the Custom Updates Publishing Tool console.
3. In the Actions pane, click Import Update(s).

4. Select Next to accept the default Single Catalog Import option.

   A wizard asks for the location of the Adobe .cab files you downloaded.

5. Select Browse to locate and select the latest Adobe Catalog for SMS.

   CUPT validates the catalog and displays the Security Warning to confirm that you would like to accept this catalog signed and published by Adobe.

6. Click Accept.

   When the import is done, the Import Software Catalog Wizard confirmation dialog box shows the number of updates imported.

7. Select Close.

8. To display Adobe software updates, click the Adobe node under Custom Updates Publishing Tool.

Publishing the Flash Player catalog

Follow these steps to publish the Flash Player catalog.

1. In the tree pane of the CUPT console, select a software name (for example, Adobe Flash Player 10) under the Adobe node.

   The result pane shows the custom update software.

2. Select the desired software version in the result pane and then select Set Publish Flag in the Actions pane. The flag should turn green.

NOTE	Initially, custom updates are not flagged in the Publish column. Each update you want to deploy must be flagged for publication. If an update is not flagged, it will not be included when the request to publish is made.

   If you want to see details about a software version, double-click it in the Result pane.

3. Select the Adobe node on the tree pane.

4. In the Actions pane, select Publish Updates.

5. Check Synchronize with Site Database of Systems Management Server and select Next.

   The Publish Wizard summary dialog box indicates the update is ready to be published.

6. Select Next to publish the update to SMS.

   When it completes, the Publish Wizard confirmation dialog box appears indicating the synchronization is successful.

7. Select Close.

   The Custom Updates Publishing Tool closes.

8. Run the SMS Administrator Console. In the console tree, select the Software Updates, select the Action menu, and click Refresh.

   The list of software updates in the details pane should contain the custom updates you published.

Confirming successful publication

Follow these steps to confirm that the catalog was successfully published.

1. In the SMS Administrator Console, navigate to the Software Updates Tree and highlight software.

   The right pane should show the same update that was published using the CUPT tool, under the type “Custom Update.”
2. In the Software Updates Tree, highlight Software Updates.
3. Navigate to the Advertisements Tree and highlight Custom Updates Tool. Right click and select Re-Run Advertisement. Select OK on the mandatory assignment pop-up note.

   Advertisement is manually initiated and Scan for Custom Updates occurs on all clients. This scan takes a period of time to complete. Forcing makes it occur immediately.

   You can view scan progress by going to System Status, Advertisement Status, Custom Updates Tool and Highlight Site in right pane. Right-click show messages and select all. This displays the current status of the Custom Update scan and install.
4. Navigate to the Reporting Tree and select Reports. Sort reports in right pane by category. Scroll down to Software Update Compliance category.
5. Select Compliance by Product Report. Leave the Product field blank and select Custom Update for the Type value.

   In the HTML report published by the Software Compliance report in this step, you should see the update and the number of machines where the update is missing or installed.

Deploying the update

Follow these steps to distribute the update across your network using SMS.

1. In the SMS Administrator Console, navigate to the Software Updates Tree and highlight Software Updates. Right-click and select distribute software updates.
2. When the wizard opens, select update type as custom update. Select SMS package as New and enter a Package Name of your choice (for example, “Adobe Flash Player Update 2”).
3. Accept the default Program Name and enter "Adobe Systems Inc." as the Organization.

4. Change Program Name to Custom Updates Tool (expedited).
5. Check all Adobe Updates that are listed. Press the Information Button to go to the Adobe website.
6. Select "I will download source files myself."
7. Select Properties and choose Import. Select the appropriate MSI file from your local hard drive for the update and click OK.
8. Check SMS Distribution Point, Collect Inventory, and Advertise. Click Browse and Select the collection to distribute to.

You should now see a program, package, and advertisement for the Update that you created. This stage can take up to 60 minutes to complete, since the client polling schedule is every 60 minutes. You can expedite this process by going to Control Panel, Systems Management, and Actions Tab on the clients. Highlight each action and click Initiate Action to trigger the client to talk to the server immediately.

To see if the update was successfully installed:

1. Navigate to the Reporting Tree and select Reports. Scroll down to Software Update Compliance category.
2. Select Compliance by Product Report. Leave the Product field blank and select Custom Update for the Type value.

In the generated report, you should see that all systems where the update was applicable are now compliant (have installed the update).

To see which systems were not able to install the update, check the software updates node of the generated report to determine Requested Systems (systems that are eligible for update) versus Compliant Systems (systems that were able to install the update).

Additional resources

The following sites provide additional information about deploying custom updates with SMS.

- Systems Management Server 2003 Concepts, Planning, and Deployment Guide at www.microsoft.com/technet/prodtechnol/sms/sms2003/cpdg
- Deploying Custom Software Updates with SMS 2003 R2 at technet.microsoft.com/en-us/magazine/cc162463.aspx

Interactive MSI installation using SMS

This section describes how to install Flash Player using the MSI installer and the Microsoft Systems Management Server (SMS) 3.0 Console. If you prefer to do a command line installation, see "Command line MSI installations" on page 53.

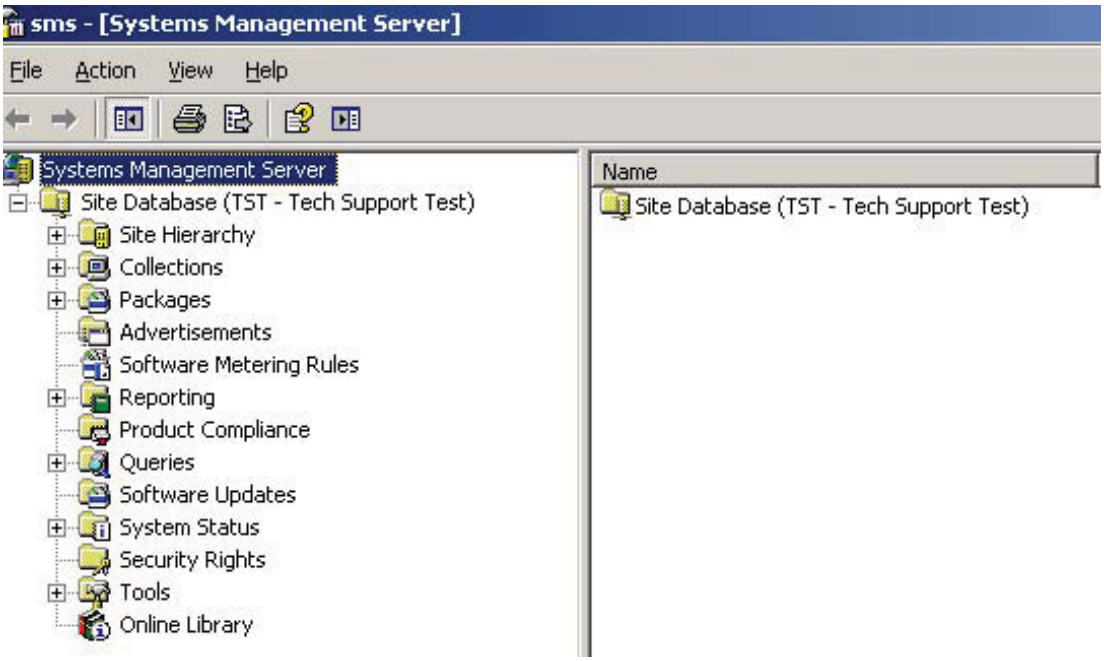
The following instructions assume the following system requirements:

- Windows 2003 Server (r2)
- SQL Server 2000 (SP4)
- SMS 2003 (SMS 3.0)
- Active Directory
- IIS (Microsoft Internet Information Server)
- BITS (Background Information Transfer)
- Flash Player MSI

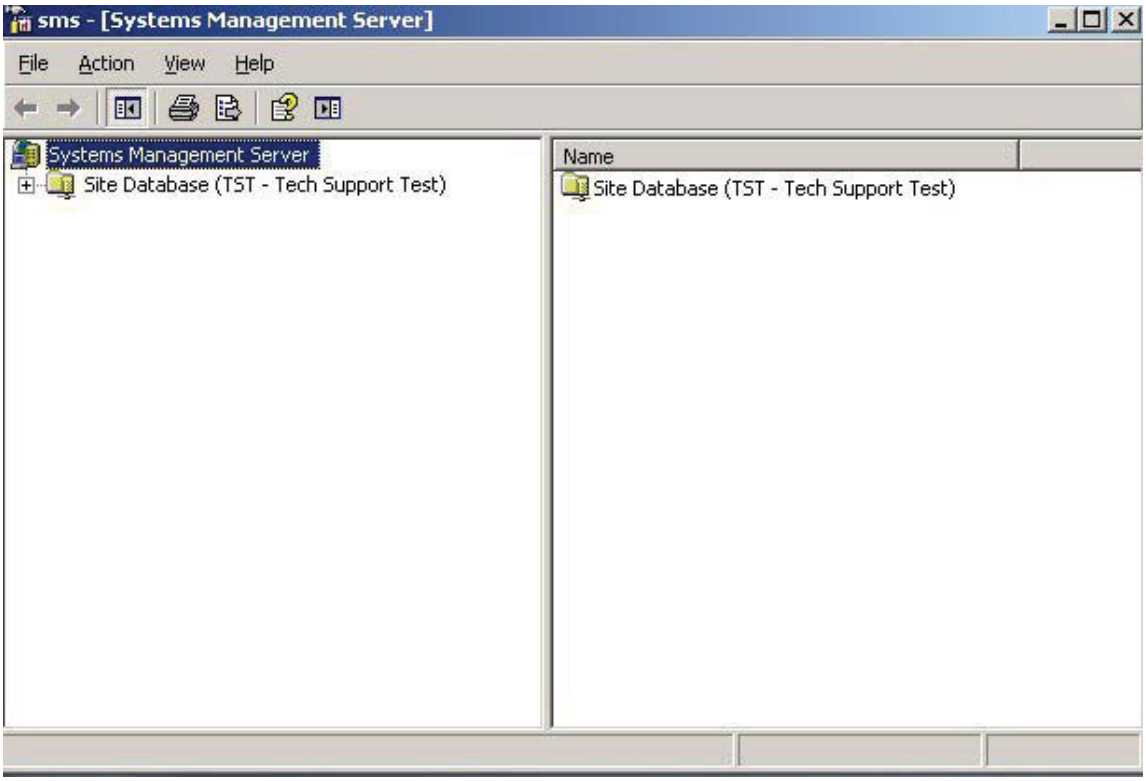
These instructions also assume that you have already installed and configured SMS 3.

NOTE	Installation using SMS can fail if the player is being installed on a machine where the logged-in user does not have administrative privileges. For information on resolving this issue, see the TechNote entitled "Flash Player MSI installation will fail on machines that don't have administrative privileges" at www.adobe.com/go/df875c9e.

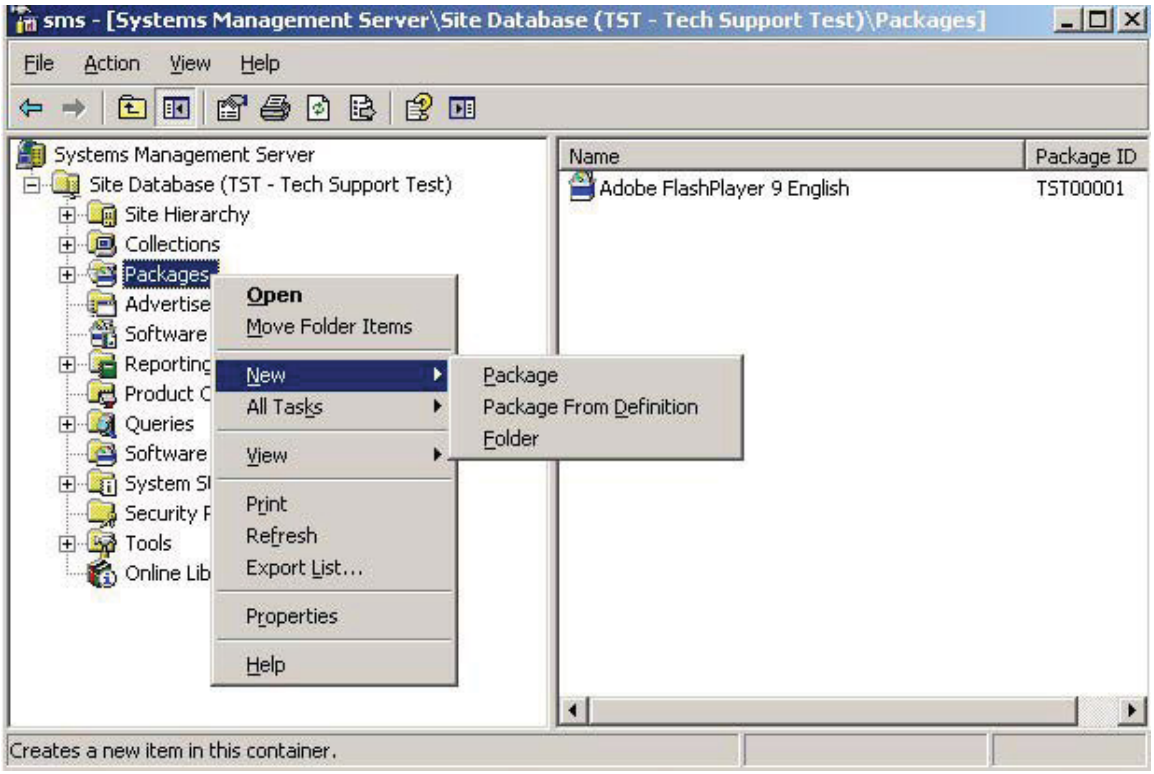
1. Start the SMS Administrator Console.



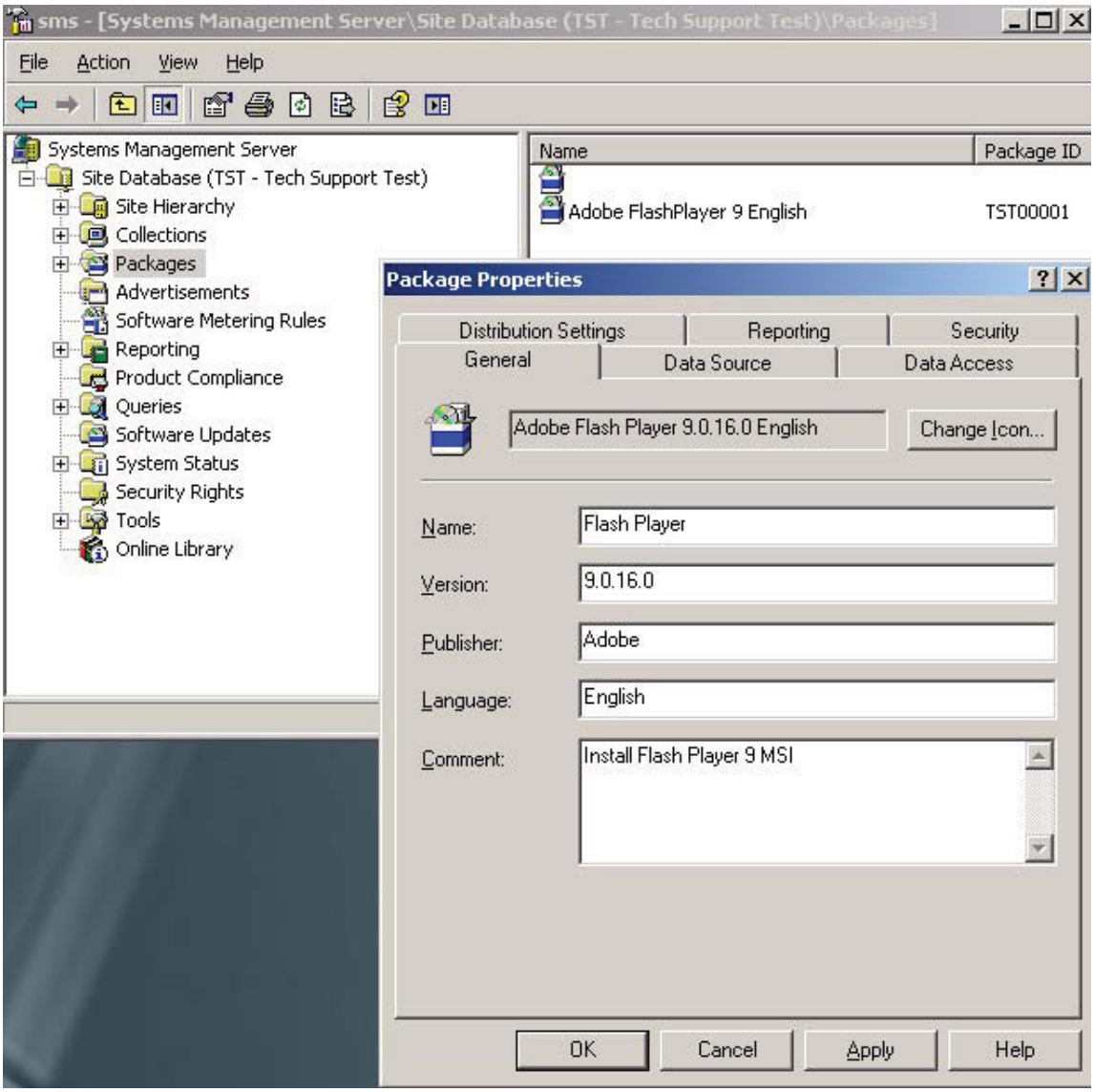
2. Expand the Site Database.



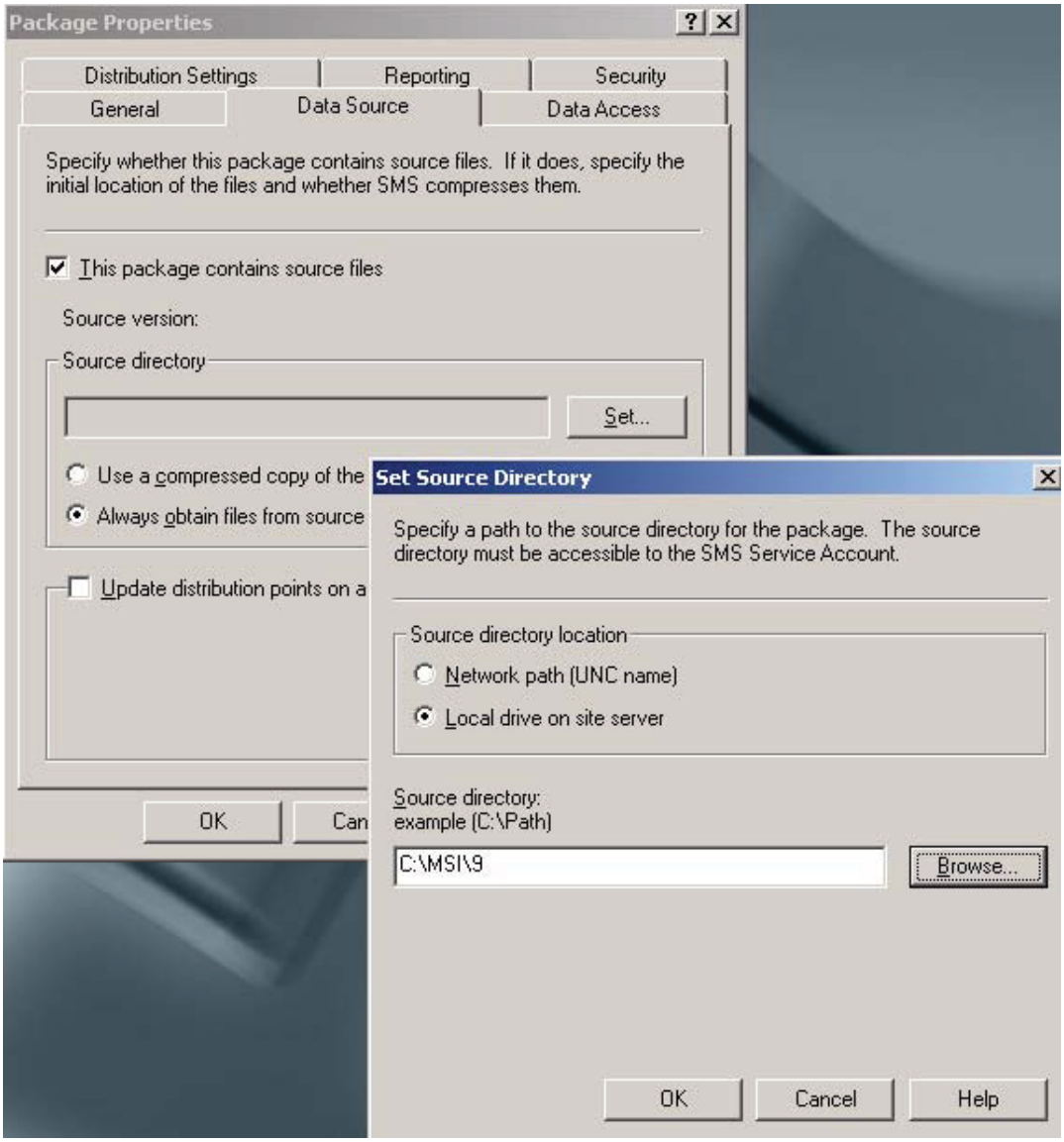
3. Right-click on Packages and select New > Package.



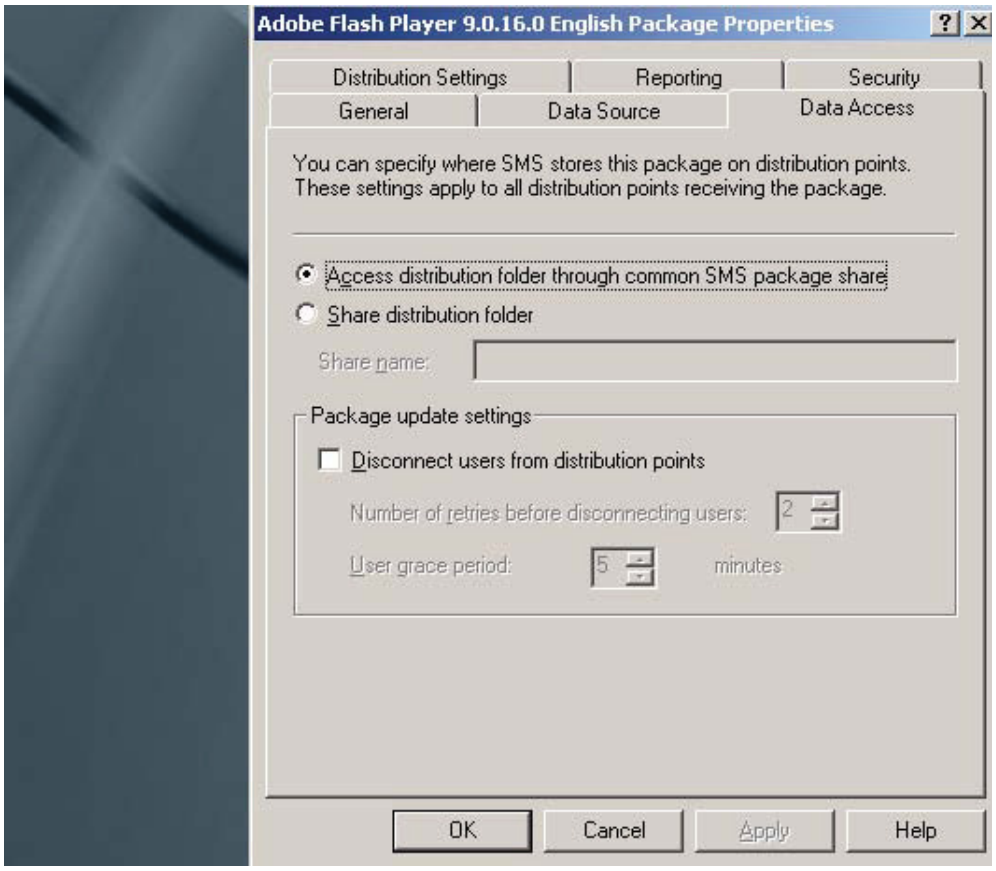
4. On the Package Properties General tab, name your package. You can also include additional data, such as the version number, publisher, language, and comments.



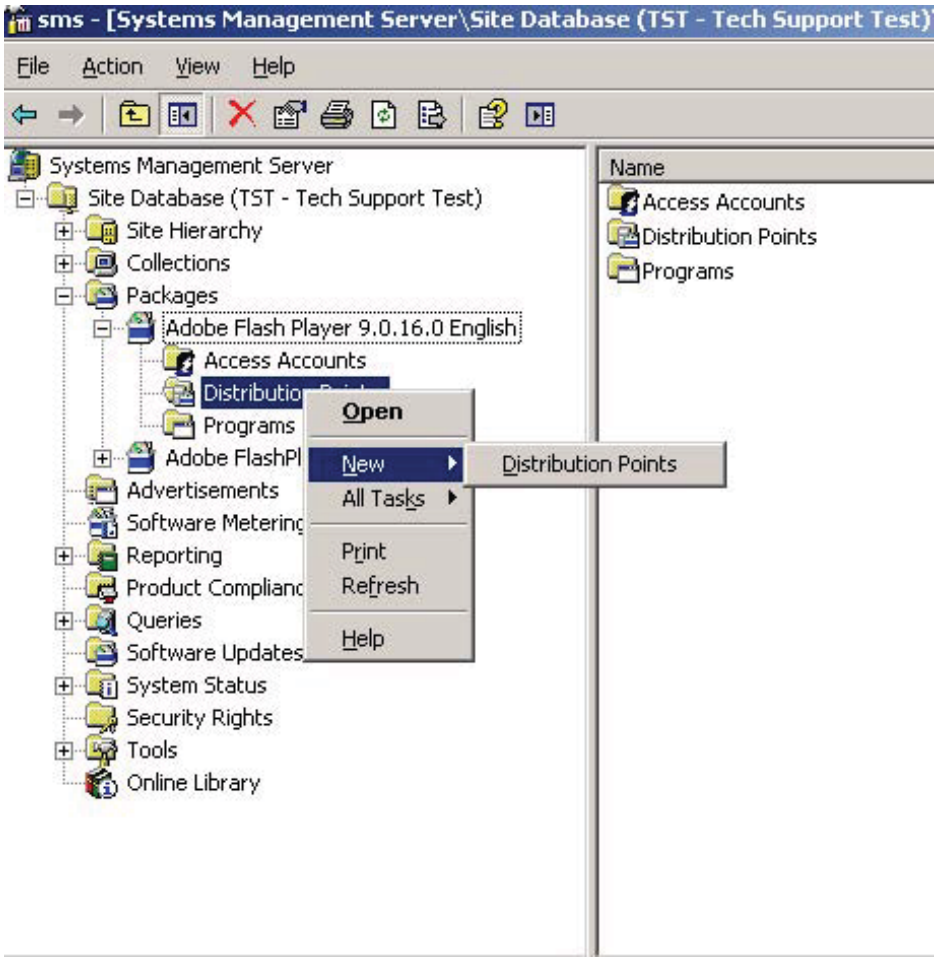
5. On the Data Source tab, enable “This package contains source files”. Click Set and browse to the network location where your source files reside. For this example, the Flash Player MSI was saved on the local C:\ drive.



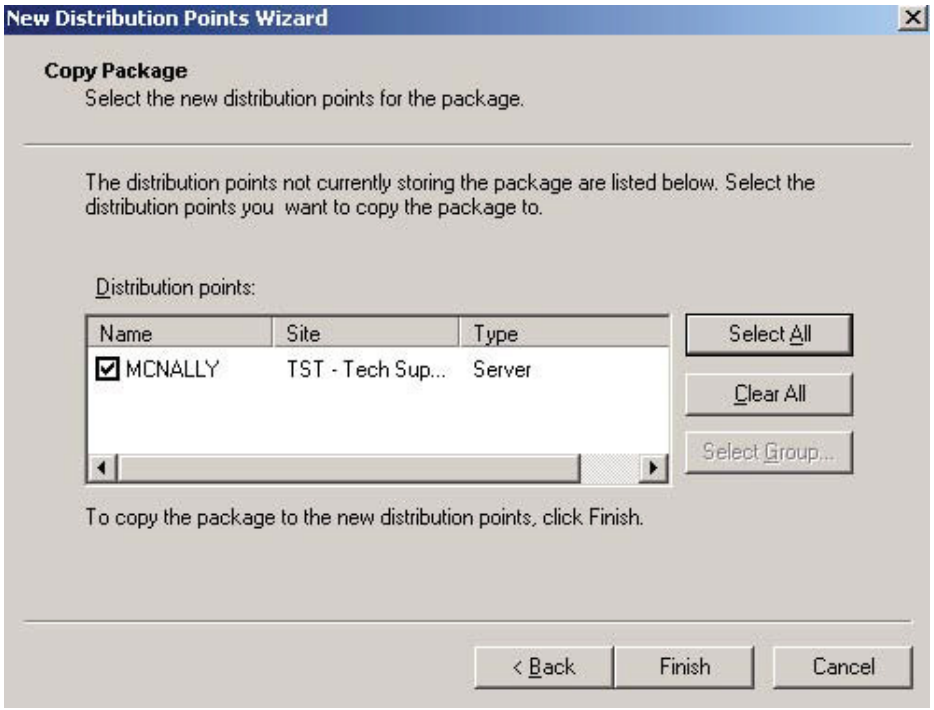
6. On the Data Access tab, select “Access distribution folder through common SMS package share” and click OK.



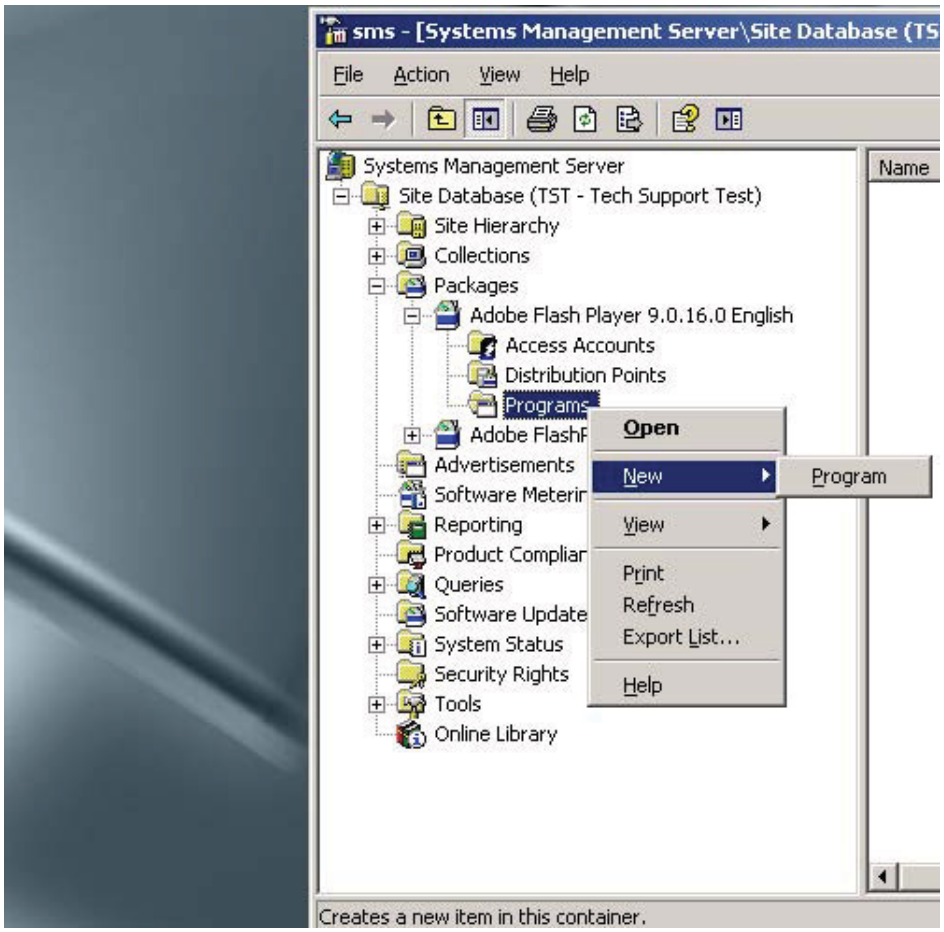
7. To make your Distribution Points (locations where SMS packages are stored), expand Packages, right-click on Distribution Points and select New > Distribution Points.



8. Select Next to start the Distribution Point wizard. Select the servers to which you want to copy the package and then click Finish.

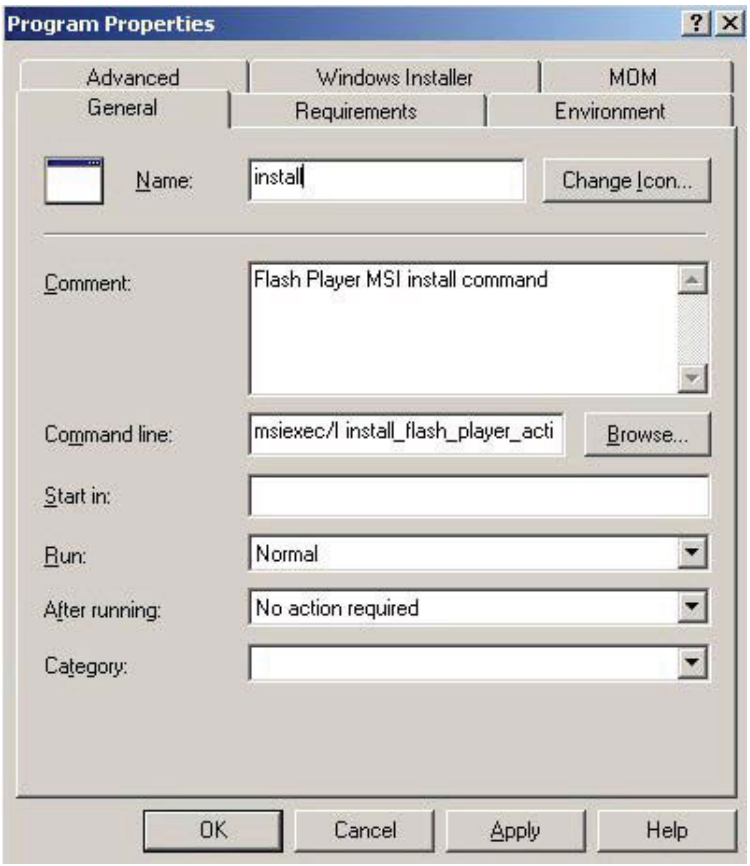


9. Right-click on Programs and select New > Program. This creates the program that will execute your deployment commands.

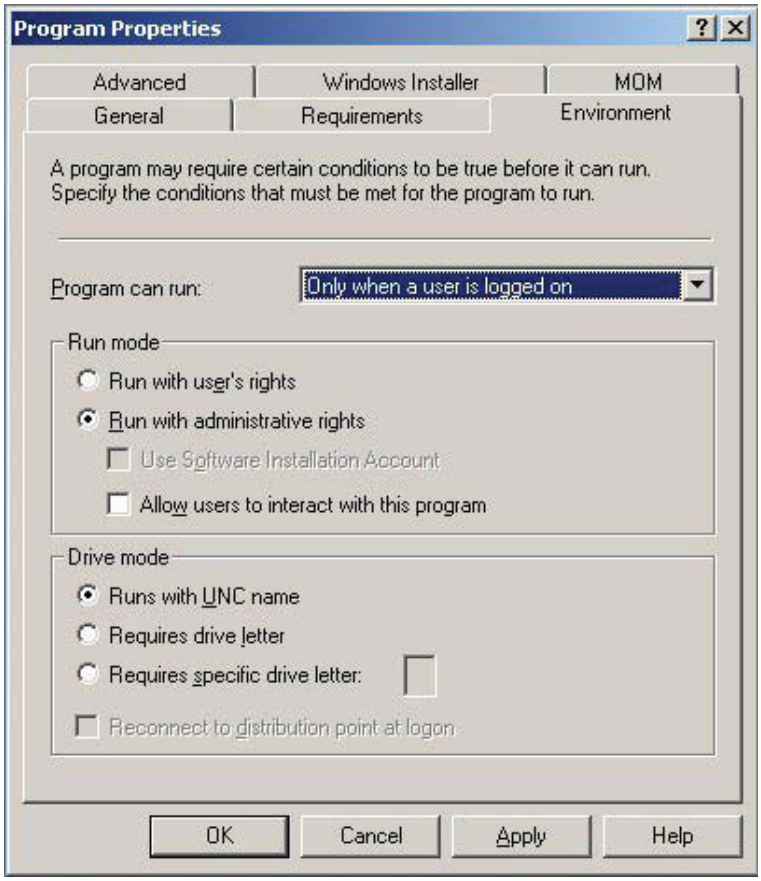


10. In the General tab, name your program and type in the command line information. In this example, we named the program "install" and then used the following command:

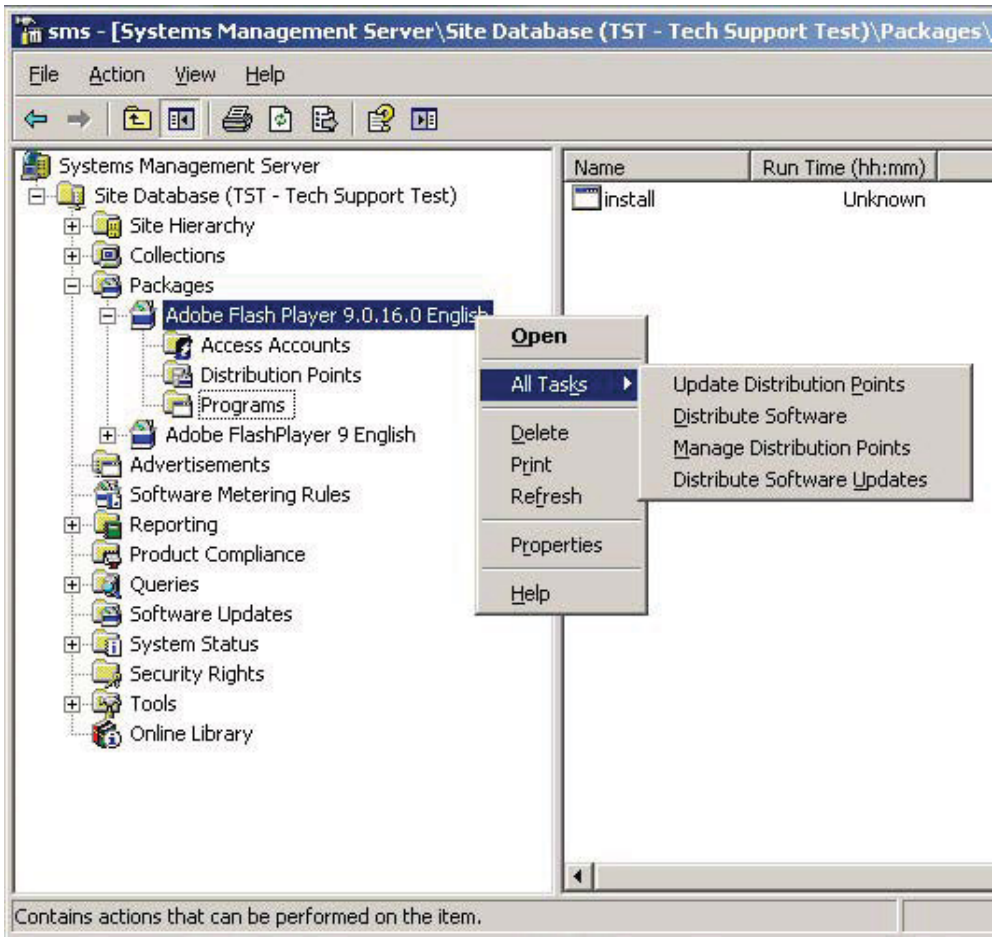
```
msiexec /i install_flash_player_active_x.msi /qn
```



11. To designate the conditions under which the application will be installed, select the Environment tab. In this example, the conditions are, “Only when a user is logged on,” “Run with administrative rights,” and “Runs with UNC name”.

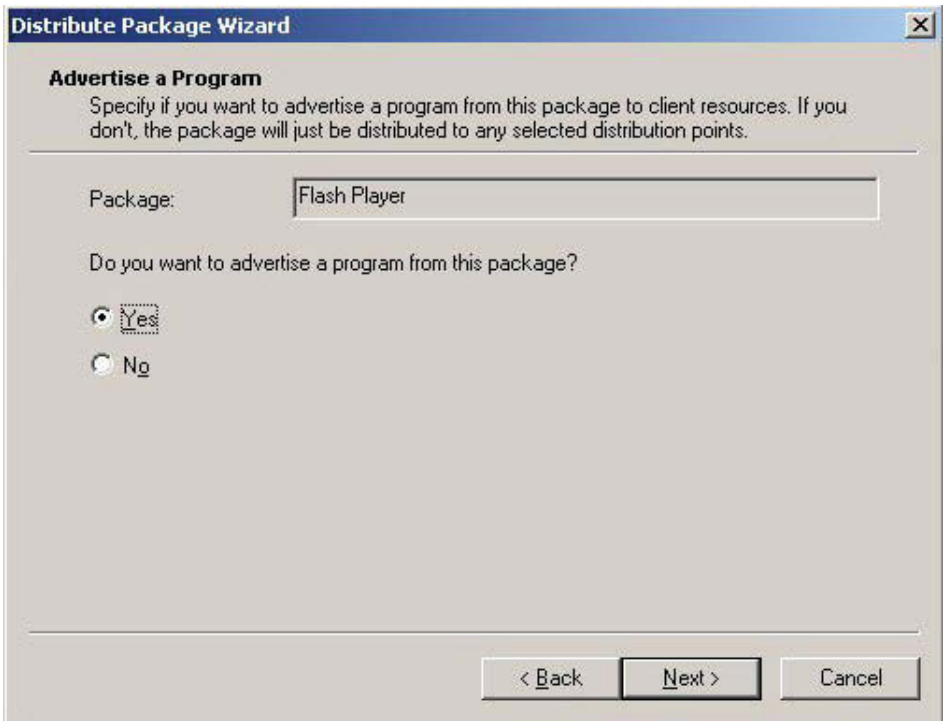


12. To make an advertisement that will apply the package program to the collection at a set time, right-click on the package and select All Tasks > Distribute Software.

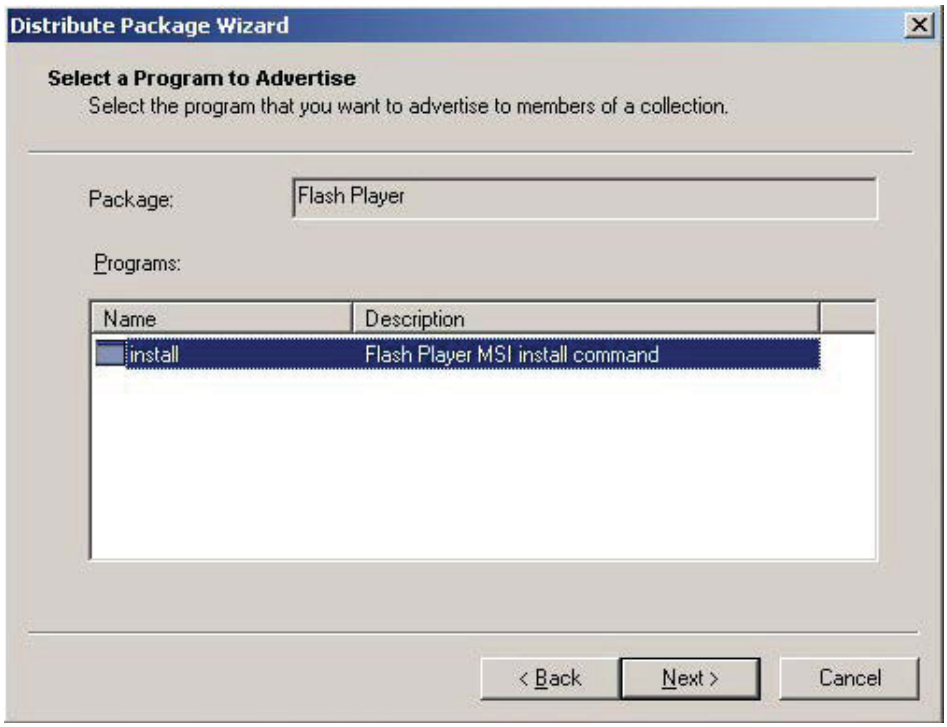


13. Select your Distribution Points and click Next.

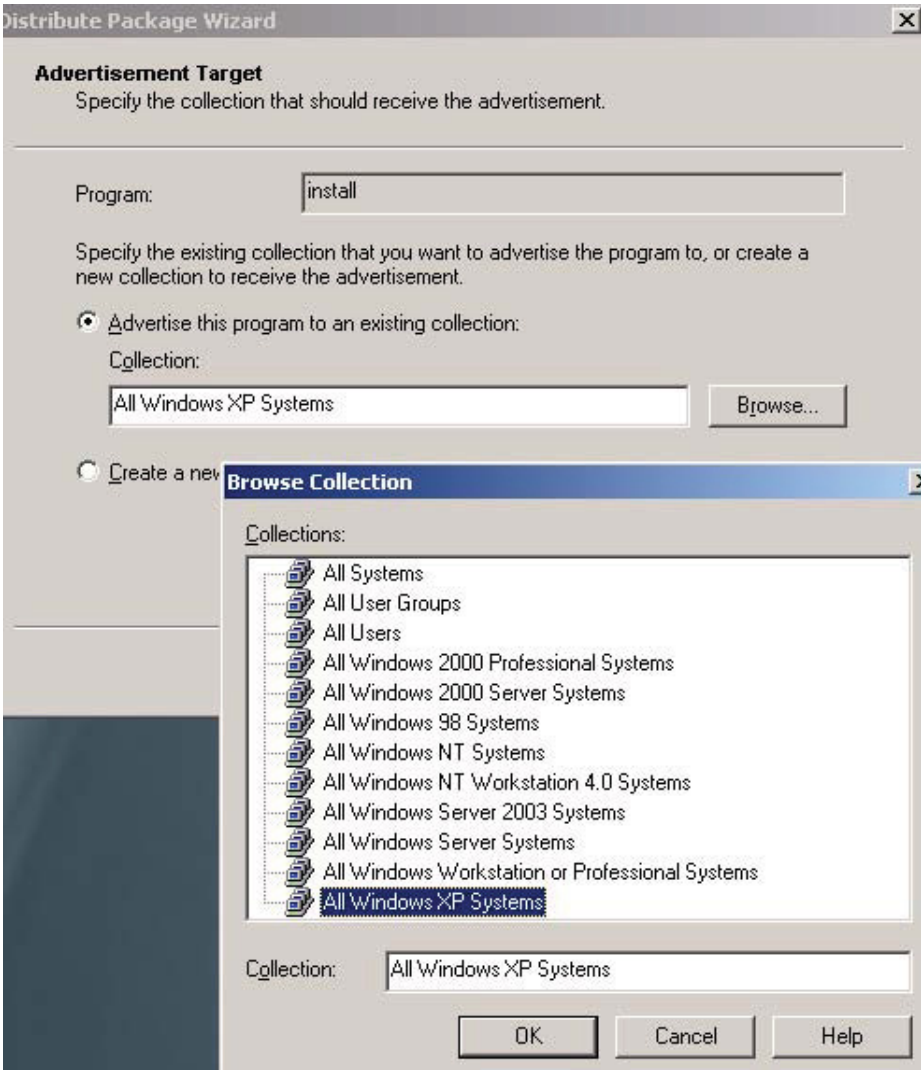
14. When asked “Do you want to advertise from this package?” choose Yes, then click Next.



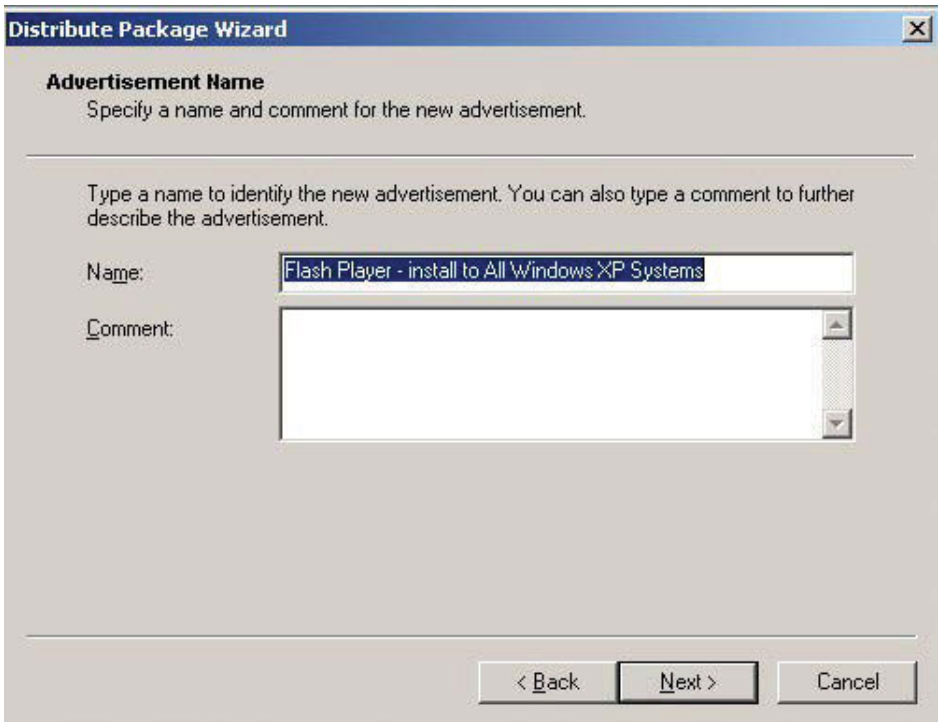
15. Select the program to advertise, then click Next. For this example, we named the program "install".



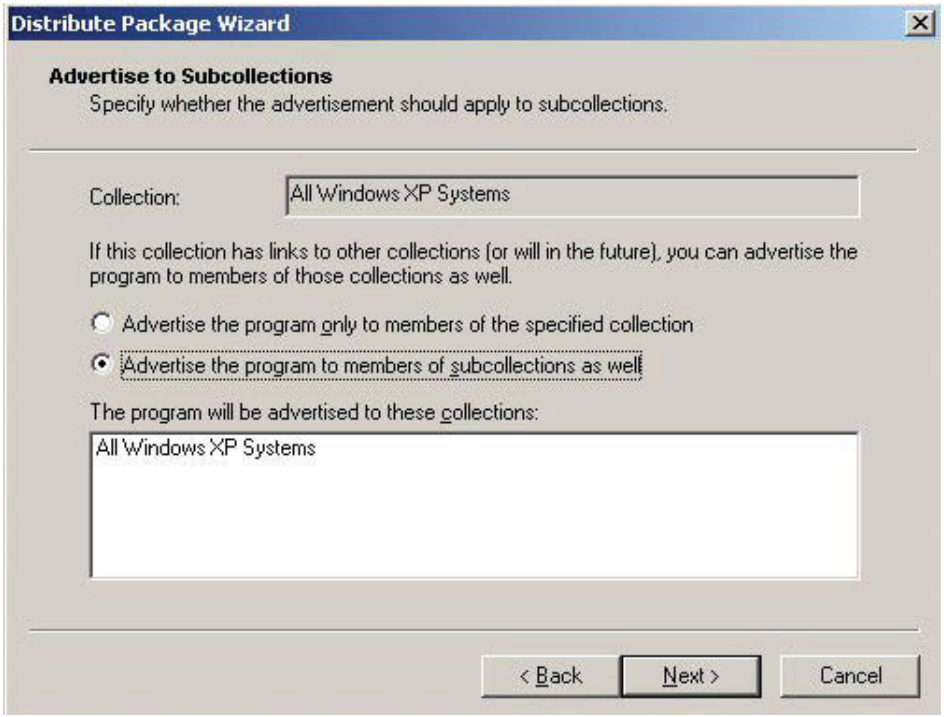
16. At this point, you can select the Collection (designated group of machines that you want to target). In the Advertisement Target pane, select, "Advertise this program to an existing collection" and select Browse. For this example, we selected "All Windows XP Systems."



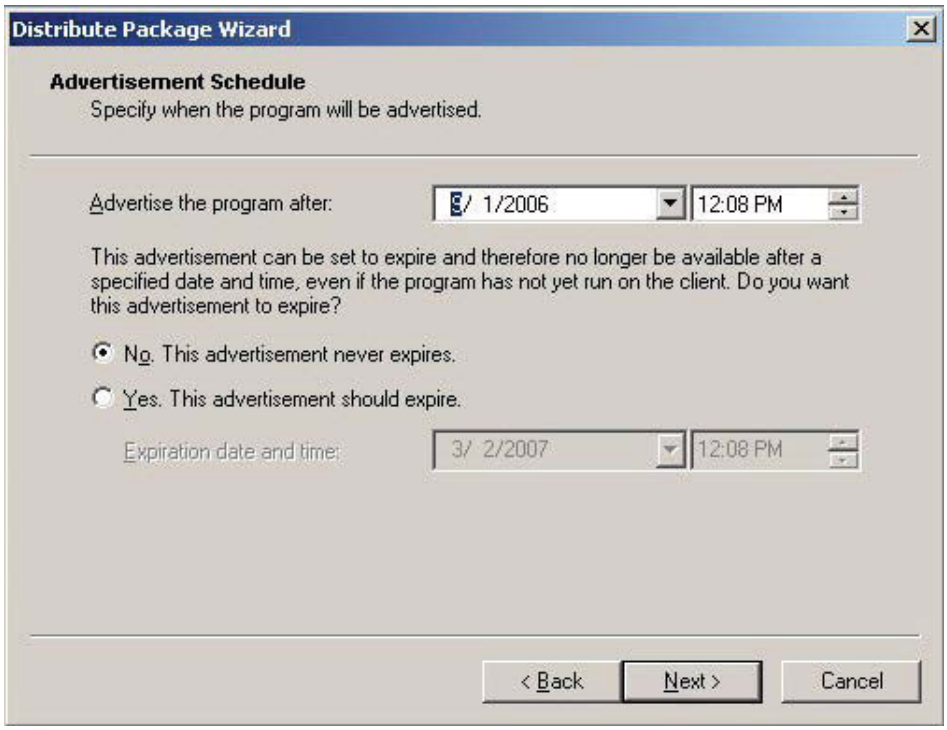
17. Select the default for the Advertisement Name, or change the name, then click Next.



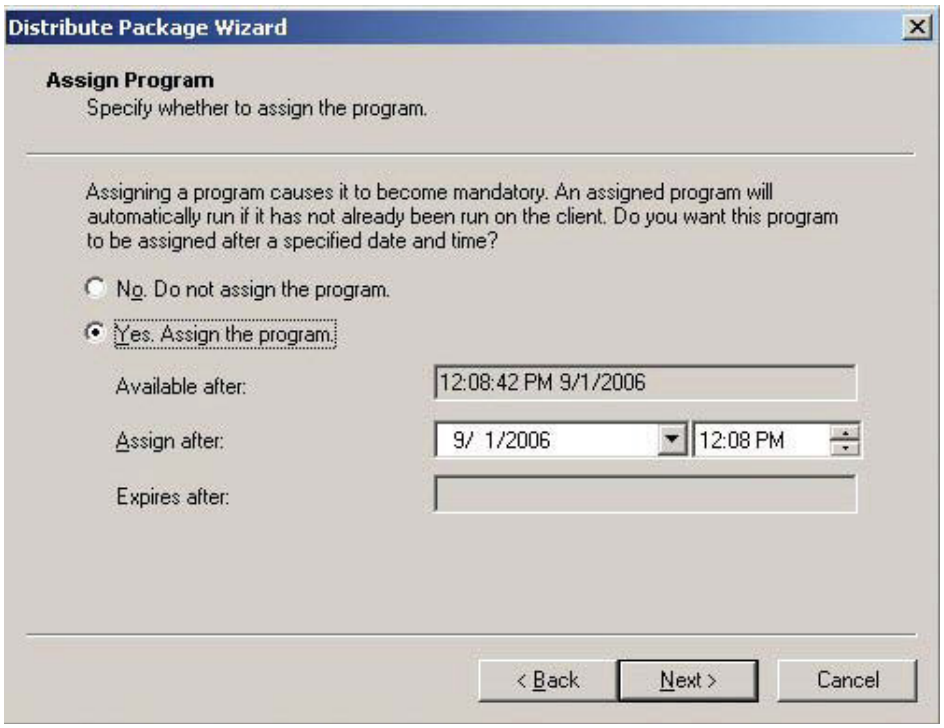
18. Specify whether the advertisement should apply to subcollections, then click Next.



19. Specify when the program will be advertised, then click Next. This allows you to advertise a program after hours when users are not on their computers.



20. You are now ready to assign your program to your collection. Select "Yes. Assign the program," then click Next



21. Look at the Details before clicking Finish.

If your deployment is successful, you will see a message that says, "Program About to Run".

Command line MSI installations

The MSI installer is provided for administrative installations using software such as Microsoft Systems Management Server (SMS). An administrative installation is the first step in preparing an MSI installer for deployment over a network. This section discusses how to deploy Flash Player over a Windows network using msiexec and the MSI installer. If you prefer to do an interactive installation using the SMS Console, see "Interactive MSI installation using SMS" on page 32.

NOTE	Installation using SMS can fail if the player is being installed on a machine where the logged-in user does not have administrative privileges. For information on resolving this issue, see the TechNote entitled "Flash Player MSI installation will fail on machines that don't have administrative privileges" at www.adobe.com/go/df875c9e.

To run an administrative installation, use the /a command line switch. For example, to run the Flash Player ActiveX control installer in interactive administrator mode, you would use this syntax:

```
msiexec /a "install_flash_player_9_activeX.msi"
```

NOTE	The examples in the rest of this chapter use the ActiveX control filename. If you are installing the browser plug-in, simply substitute the correct filename in your installation.

On some machine configurations, spaces in the MSI filename interfere with running the installer from the command line, even with quotes around it. If you rename the MSI file for any reason, do not use any spaces in the filename.

When started as shown above, the installer runs though its AdminUISequence, involving a series of dialog boxes. The first dialog box is a simple welcome screen, and the next dialog prompts for the Network location that you want to install to.

Clicking Next in the Welcome dialog runs the Network Location dialog. Clicking Install in this dialog box deploy the admin tree to a network share.

NOTE	The admin install includes only those files contained within the MSI file itself. Other support files required by the installation such as bootstrap files, MSI runtime installers, or patches, should be copied to the shared folder by some other means of your choice (manually, with a script, batch file, and so on).

Once the admin install is deployed to the shared folder, there are different ways that it can be used, in turn, to install the product onto a workstation. These are discussed in the rest of this section.

Manually launch the installer on the client

One easy way to pull the installation from an administrative image is to run it manually, by sitting at the client machine and launching it interactively from the site on which it is being shared. You could do this either by double-clicking the bootstrap file, or by double-clicking the MSI file. The bootstrap file is the recommended one to use, as it automatically installs the required version of the MSI runtime first, if needed, before launching the MSI file in turn.

NOTE	If you've renamed the MSI file to avoid command line problems with spaces in the filename, the bootstrap file will no longer work, because the bootstrap file is looking for a specific hard-coded filename. In this case, run the MSI file directly instead.

Launch the installer on the client using quiet mode

If you don't need to customize the installation options, then you can run the installation non-interactively. This method requires with a command line switch, as shown below. When run in this mode, the default options are used for all items that would be presented as choices in the interactive install.

```
msiexec /i "install_flash_player_9_activeX.msi" /qn
```

The simple command line syntax shown above works in most cases, but other command line elements and switches are available. A more comprehensive version of the syntax looks like this (to be entered all on one line):

```
%Comspec% /c msiexec /i
  "\\network path\install_flash_player_9_activeX.msi" /qn
```

In both cases, the final `/qn` switch must be on the same line as the rest of the command.

The arguments used in the command line example above are described below.

- `%Comspec%` is an environment variable provided by Windows. It points to the command interpreter, cmd.exe.
- `/c` is a switch passed to cmd.exe telling the shell to wait until the msiexec.exe command completes before proceeding. Without this switch, the shell will execute subsequent commands before the current command finishes.
- `msiexec.exe` is the Windows installer runtime. When you double-click an MSI file (for example, foo.msi) you are implicitly running **msiexec /i foo.msi**.
- `/i` instructs MSIEXEC to install the MSI file listed after the switch. There is also an `/x` switch that uninstalls the MSI file specified after the `/x` switch.
- `/qn` specifies a user interface level for the action. The `/qn` switch suppresses all prompts and is therefore useful for silent installations. When attempting to debug, you can switch to `/qb`, which displays basic modal dialogs.

For more information about command line options available for msiexec, see "Command-Line Options" in the MSDN Library at msdn.microsoft.com/en-us/library/aa367988.aspx.

Reinstalling a Flash Player using a batch routine

If you need to uninstall and reinstall the Flash Player, you can use a batch file like this one:

```
REM Begin quietInstall.bat
REM Uninstall Flash Player ActiveX
%Comspec% /c msiexec /x "\\network path\install_flash_player_9_activeX.msi"
  /qn
REM Install Flash Player ActiveX
%Comspec% /c msiexec /i "\\network path\install_flash_player_9_activeX.msi"
  /qn
REM End quietInstall.bat
```

Windows registry keys

In addition to the registry keys you can use to determine the installed version of a player (see "Player versions" on page 14), Flash Player creates other registry keys when it is installed or registered. These keys are summarized in the Flash Player TechNote entitled "Registry permissions required for Flash Player install or update" at www.adobe.com/go/tn_19148.

DMG installation for Macintosh

For Mac OS X Universal Binary or Power PC, you use a DMG installer. Double-click the DMG file to mount it on your desktop and create an Adobe Flash Player Package installer. Double-click this application and follow the guided installation instructions.

Customizing player behavior

After you deploy the player, you can install a privacy and security configuration file (mms.cfg) to specify rules about Flash Player security options and Flash application access to the file system and network. The file controls security-related behavior of the player after installation.

The primary purpose for the mms.cfg file is to support the corporate and enterprise environments where the IT department would like to install Flash Player across the enterprise, while enforcing some common global security and privacy settings (supported with installation-time configuration choices). The mms.cfg file can be used to control data loading operations, user privacy, auto-update behavior, and local file security.

For detailed information about customizing player behavior, see Chapter 3, "Administrator Settings," on page 59.

Troubleshooting installation problems

The following TechNotes address installation problems you may encounter.

- Troubleshoot Adobe Flash Player installation for Windows (www.adobe.com/go/tn_19166)
- Troubleshoot Adobe Flash Player for Intel-based Macs (www.adobe.com/go/2dda3d81)
- Troubleshooting Adobe Flash Player for Linux and Solaris (www.adobe.com/go/tn_15397)

Additional resources

For answers to questions regarding Flash Player licensing and deployment, see Adobe Player Licensing at www.adobe.com/licensing/distribution and the player Distribution FAQ at www.adobe.com/licensing/distribution/faq.

The following sites outside Adobe provide general information on deploying software on Windows systems.

- Windows Installer Resources for System Administrators at www.installsite.org/pages/en/msi/admins.htm.
- Applying Small Updates by Patching an Administrative Image in the MSDN library at msdn.microsoft.com/en-us/library/aa367573.aspx.
- Applying Small Updates by Reinstalling the Product in the MSDN library at msdn.microsoft.com/en-us/library/aa367575.aspx.
- For information on detecting player version from a web site, see the “Detection and Installation” section at the Flash Player Developer Center (www.adobe.com/devnet/flashplayer/detection_installation.html).

CHAPTER 3

Administrator Settings

3

This chapter describes ways you can create and place files on the end user's machine to manage features related to security, privacy, use of disk space, and so on.

This chapter includes the following sections:

Privacy and security settings (mms.cfg) 59
The Global FlashPlayerTrust directory 74

Privacy and security settings (mms.cfg)

As a network administrator, you can install Flash Player across the enterprise while enforcing some common global security and privacy settings (supported with installation-time configuration choices). To do this, you install a file named mms.cfg on each client machine.

The mms.cfg file is a text file. When Flash Player starts, it reads its settings from this file, and uses them to manage functionality as described in the following sections.

What's new

The following mms.cfg options are new in Flash Player 10.0.2.

- OverrideGPUValidation
- ProductDisabled
- RTMFPP2PDisable
- RTMFPTURNProxy

mms.cfg file location

Assuming a default Windows installation, Flash Player looks for the mms.cfg file in the following system directory:

- **Windows (Vista, XP and 2000)** %WINDIR%\System32\Macromed\Flash

> **NOTE** The %WINDIR% location represents the Windows system directory, such as C:\WINDOWS.

- **Macintosh** /Library/Application Support/Macromedia
- **Linux** /etc/adobe/

> **NOTE** Unlike Windows and Macintosh, the Linux player is in a directory named adobe, not in one named Macromed or Macromedia.

You might use third-party administration tools, such as Microsoft System Management Server, to replicate the configuration file to the user's computer.

Use the standard techniques provided by your operating system to hide or otherwise prevent end users from seeing or modifying the mms.cfg file on their systems.

Setting options in the mms.cfg file

This section discusses how to format and set options in the mms.cfg file. The value of some mms.cfg options can be queried through the use of ActionScript. When this is possible, the ActionScript API is noted in the option's description.

File format

The format of the mms.cfg file is a series of `name = value` pairs separated by carriage returns. If a parameter is not set in the file, Flash Player either assumes a default value or lets the user specify the setting by responding to pop-up questions, or by using Settings dialog boxes or the Settings Manager. (For more information on how the user can specify values for certain options, see Chapter 4, "User-Configured Settings," on page 77.)

The options in the mms.cfg file use the following syntax:

```
ParameterName = ParameterValue
```

Only one option per line is supported. Specify Boolean parameters either as `"true"` or `"false"`, or as `1` or `0`, or as `"yes"` or `"no"`.

Comments are allowed. They start with a # symbol and go to the end of the line. This symbol can be used to insert comments or to temporarily disable directives.

Whitespace is allowed, including blank lines or spaces around equal signs (=).

Character encoding

Some mms.cfg directives may have values that include non-ASCII characters, so the character encoding of the file is significant in those cases. We support a standard text file convention: the file may use either UTF-8 or UTF-16 Unicode encoding, either of which must be indicated by including a "byte order mark" (BOM) character at the beginning of the file; if no BOM is found, Flash Player assumes that the file is encoded using the current system default code page. Many popular text editors, including Windows Notepad and Mac TextEdit, are capable of writing UTF-8 or UTF-16 files with BOMs, although you may need to specify that as an option when saving.

Summary of mms.cfg options

The following table summarizes the options available in mms.cfg, in alphabetical order.

Option	Description
AllowUserLocalTrust	Lets you prevent users from designating any files on local file systems as trusted.
AssetCacheSize	Lets you specify a hard limit, in MB, on the amount of local storage that Flash Player uses for the storage of common Flash components.
AutoUpdateDisable	Lets you prevent Flash Player from automatically checking for and installing updated versions.
AutoUpdateInterval	Lets you specify how often to check for an updated version of Flash Player.
AVHardwareDisable	Lets you prevent SWF files from accessing webcams or microphones.
DisableDeviceFontEnumeration	Lets you prevent information on installed fonts from being displayed.
DisableNetworkAndFilesystemIn HostApp	Lets you prevent networking or file system access of any kind.
DisableProductDownload	Lets you prevent native code applications that are digitally signed and delivered by Adobe from being downloaded.
DisableSockets	Lets you enable or disable the use of the `Socket.connect()` and `XMLSocket.connect()` methods.
EnableSocketsTo	Lets you create a whitelist of servers to which socket connections are allowed.

Option	Description
EnforceLocalSecurityInActiveXHostApp	Lets you enforce local security rules for a specified application.
FileDownloadDisable	Lets you prevent the ActionScript FileReference API from performing file downloads.
FileUploadDisable	Lets you prevent the ActionScript FileReference API from performing file uploads.
FullScreenDisable	Lets you disable SWF files playing via a browser plug-in from being displayed in full-screen mode.
LegacyDomainMatching	Lets you specify whether SWF files produced for Flash Player 6 and earlier can execute an operation that has been restricted in a newer version of Flash Player.
LocalFileLegacyAction	Lets you specify how Flash Player determines whether to execute certain local SWF files that were originally produced for Flash Player 7 and earlier.
LocalFileReadDisable	Lets you prevent local SWF files from having read access to files on local hard drives.
LocalStorageLimit	Lets you specify a hard limit on the amount of local storage that Flash Player uses (per domain) for persistent shared objects.
OverrideGPUValidation	Overrides validation of the requirements needed to implement GPU compositing.
ProductDisabled	Creates a list of ProductManager applications that users are not permitted to install or launch.
RTMFPP2PDisable	Specifies how the NetStream constructor connects to a server when a value is specified for `peerID`, the second parameter passed to the constructor.
RTMFPTURNProxy	Lets Flash Player make RTMFP connections through the specified TURN server in addition to normal UDP sockets.
ThirdPartyStorage	Lets you specify whether third-party SWF files can read and write locally persistent shared objects.

This document describes mms.cfg options that let you do the following:

- Control access to camera, microphone, and system font information (see "Privacy options" on page 63).
- Specify whether SWF files playing in a browser can be displayed in full-screen mode (see "User interface option" on page 64).
- Control access to the local file system (see "Data loading and storage options" on page 64).
- Specify settings for Flash Player auto-update (see "Update options" on page 67).
- Specify adjustments to Flash Player's default security model (see "Security options" on page 69).
- Specify whether low-level socket connections are allowed (see "Socket connection options" on page 72).
- Override settings related to GPU compositing (see "GPU Compositing" on page 72).
- Specify settings related to Peer-to-Peer connections using the RTMFP protocol (see "RTMFP options" on page 73).

Where a setting has a default value, it is displayed in **bold type**.

Privacy options

Settings in this category let you: disable the use of camera and microphone devices to capture video and audio streams; and disable the ability to view the list of system fonts installed on a user's computer.

AVHardwareDisable

`AVHardwareDisable = [ 0, 1 ]` (**0 = false**, 1 = true)

If this value is set to 1, SWF files cannot access webcams or microphones. If this value is 0 (the default), the Settings Manager or Settings tabs let the user specify settings for access to webcams and microphones. (See "Privacy options" on page 78.)

If this value is set to 1, the privacy pop-up dialog never appears. However, the user can still access the Privacy tab and the Settings Manager, as well as tabs to let them designate which camera or microphone an application can use. These settings appear functional, but any choices the user makes are ignored. Also the recording level meter on the Microphone tab is disabled, and the Camera tab does not bring up a thumbnail of what the camera is seeing.

NOTE

In ActionScript, an author can query the `System.capabilities.avHardwareDisable` property to determine the value of this setting.

DisableDeviceFontEnumeration

`DisableDeviceFontEnumeration = [ 0, 1 ]` (**0 = false**, 1 = true)

This setting controls whether the `Font.enumerateFonts()` method in ActionScript 3.0 and the `TextField.getFontList()` method in ActionScript 1.0 and 2.0 return the list of fonts installed on a user's system. If this value is 1, information on installed fonts cannot be returned. If this value is 0 (the default), information on installed fonts can be returned.

User interface option

The setting in this category determines whether SWF files playing in a browser can be displayed in full-screen mode.

FullScreenDisable

`FullScreenDisable = [ 0, 1 ]` (**0 = false**, 1 = true)

This setting controls whether a SWF file playing via a browser plug-in can be displayed in full-screen mode; that is, taking up the entire screen and thus obscuring all application windows and system controls. If you set this value to 1, SWF files that attempt to play in full-screen mode fail silently. The default value is 0.

Full-screen mode is implemented with a number of security options already built in, so you might choose to disable it only in specific circumstances. To learn more about full-screen mode, see www.adobe.com/go/fullscreen.

Data loading and storage options

Settings in this category let you do the following:

- prevent local SWF files from reading local files
- prevent uploading and downloading of files between remote servers and local file systems
- limit (optionally to zero) the amount of local storage web sites can use for persistent shared objects
- limit (optionally to zero) the size of the asset cache (also called the cross-domain cache)
- prevent third-party SWF files from reading and writing locally persistent shared objects

NOTE	Disabling features may cause certain web sites and applications to work incorrectly. If these features are needed for applications running in your environment, do not disable them.

LocalFileReadDisable

`LocalFileReadDisable = [ 0, 1 ]` (**0 = false**, 1 = true)

Setting this option to 1 prevents local SWF files from having read access to files on local hard drives; that is, local SWF files can't even run. In addition, remote SWF files are unable to upload or download files. The default value is 0.

If this value is set to 1, ActionScript cannot read any files referenced by a path (including the first SWF file that Flash Player opens) on the user's hard disk. Any ActionScript API that loads files from the local file system is blocked. File upload/download via methods of the FileReference and FileReferenceList ActionScript APIs are also blocked if this flag is set. In addition, any values set for FileDownloadDisable and FileUploadDisable are ignored.

It is important to remember that, except for uploading and downloading files, the only SWF files that can read local files are SWF files that are themselves local. Therefore, you do not need to use this option to prevent remote SWFs from reading local data; that is always prevented anyway.

If this option is disabled, the ActionScript methods `FileReference.browse()` and `FileReferenceList.browse()` are also disabled.

NOTE	In ActionScript 1.0 and 2.0, an author can use the `System.capabilities.localFileReadDisable` API to query the value of this setting. The corresponding ActionScript 3.0 API is `Capabilities.localFileReadDisable`.

FileDownloadDisable

`FileDownloadDisable = [ 0, 1 ]` (**0 = false**, 1 = true)

If this value is set to 1, the ActionScript `FileReference.download()` method is disabled; the user is not prompted to allow a download, and no downloads using the FileReference API are allowed. If this value is set to 0 (the default), Flash Player allows the ActionScript `FileReference.download()` method to ask the user where a file can be downloaded to, and then Flash Player downloads the file after the user approves the file save location. Files are never downloaded without user approval.

FileUploadDisable

`FileUploadDisable = [ 0, 1 ]` (**0 = false**, 1 = true)

If this value is set to 1, all `FileReference.upload()`, `FileReference.browse()`, and `FileReferenceList.browse()` activity is disabled; the user is not prompted to upload files, and no uploads using the FileReference API are allowed. If this value is set to 0 (the default), Flash Player allows files to be uploaded using the FileReference API. The user is prompted to select a file to upload and to approve the selection. Files are never uploaded without user approval.

LocalStorageLimit

`LocalStorageLimit = [ 1, 2, 3, 4, 5, 6 ]` (1 = no storage, 2 = 10 KB, 3 = 100 KB, 4 = 1 MB, 5 = 10 MB, **6 = user specifies upper limit**)

This value specifies a hard limit on the amount of local storage that Flash Player uses (per domain) for persistent shared objects. The user can use the Settings Manager or Local Storage Settings dialog box to specify local storage limits (see "Local storage options" on page 79). If no value is set here and the user doesn't specify storage limits, the default limit is 100 KB per domain. If this value is set to 6 (the default), the user specifies the storage limits for each domain.

If LocalStorageLimit is set, the Local Storage tab shows the limit specified. and the user can use this tab as if the limit does not exist. If the user sets more restrictive settings than the value set by LocalStorageLimit, they are honored (and displayed the next time the Settings dialog box is loaded). However, if the user selects settings higher than the limit set by LocalStorageLimit, the user's settings are ignored.

The local file storage limit is best obtained from the Settings dialog box, because this security setting is just a maximum value, and the user may have set a lower limit.

ThirdPartyStorage

`ThirdPartyStorage = [ 0, 1 ]` (0 = false, 1 = true)

Third party refers to SWF files that are executing within a browser and have an originating domain that does not match the URL displayed in the browser window.

If this value is set to 1, third-party SWF files can read and write locally persistent shared objects. If this value is set to 0, third-party SWF files cannot read or write locally persistent shared objects.

This setting does not have a default value. If it is not included in the mms.cfg file, the Settings Manager or Local Storage Settings dialog box lets the user specify whether to permit locally persistent shared objects. If the user doesn't make any changes, the default is to permit shared objects. For more information on third-party shared objects, see the article entitled "What are third-party local shared objects?" at www.adobe.com/products/flashplayer/articles/thirdpartylso.)

AssetCacheSize

```
AssetCacheSize = [ 0, number of megabytes ]
```

This value specifies a hard limit, in MB, on the amount of local storage that Flash Player uses for the storage of common Flash components. If this option is not included in the mms.cfg file, the Settings Manager lets the user specify whether to permit component storage. However, the user can't specify how much local storage space to use. The default limit is 20 MB.

Setting this value to 0 disables component storage, and any components that have already been downloaded are purged the next time Flash Player runs.

Update options

Settings in this category let you configure the auto-update mechanism used by Flash Player. You can increase or decrease the frequency of checks for newer versions, or disable auto-update entirely.

Flash Player supports notification of software updates by periodically checking for new versions of the player on the adobe.com site. Flash Player never runs in the background to perform the auto-update check. This anonymous check is only performed when the player is loaded to view Flash content, typically in the browser, and by default only occurs if it has been at least 30 days since the last time it checked for updates.

The auto-update notification settings can be configured by users or by options in the mms.cfg file. Users can set the frequency of the checks or disable auto-update notification by using the Global Notifications Settings Panel in the Flash Player Settings Manager.

If you want to enforce standardized update settings for all users, you can use the mms.cfg options discussed in this section.

AutoUpdateDisable

`AutoUpdateDisable = [ 0, 1 ]` (**0 = false**, 1 = true)

If this value is set to 0 (the default), Flash Player lets the user enable or disable auto-update in the Settings Manager. If this value is set to 1, Flash Player disables auto-update, which prevents Flash Player from automatically checking for and installing updated versions. You can't use this option to prevent the user from disabling auto-update.

NOTE	If this value is set to 1, or if the user disables auto-update, the remaining options in this section are ignored.

AutoUpdateInterval

`AutoUpdateInterval = [ number of days  ]`

If this is a negative value (the default), Flash Player uses the auto-update interval value specified in the Settings Manager. (If users don't make any changes with the Settings Manager, the default is every 30 days.) If this value is set to 0, Flash Player checks for an update every time it starts. If this is a positive value, the value specifies the minimum number of days between update checks.

DisableProductDownload

`DisableProductDownload = [ 0, 1 ]` (**0 = false**, 1 = true)

If this value is set to 0 (the default), Flash Player can install native code applications that are digitally signed and delivered by Adobe. Adobe uses this capability to deliver Flash Player updates through the developer-initiated Express Install process, and to deliver the Adobe Acrobat Connect screen-sharing functionality. If this value is set to 1, these capabilities are disabled.

However, if you want to enable some but not all product downloads, set this value to 0 (or omit it) and then use the ProductDisabled option to specify which product downloads are not permitted.

ProductDisabled

`ProductDisabled = application name`

This option is effective only when DisableProductDownload has a value of 0 or is not present in the mms.cfg file; it creates a list of ProductManager applications that users are not permitted to install or launch. Unlike most other mms.cfg options, you can use this option as many times as is appropriate for your environment.

Security options

These options let you modify the default Flash Player security model. For more information on the security model, see Chapter 5, "Security Considerations."

LegacyDomainMatching

`LegacyDomainMatching = [ 0, 1 ]` (0 = false, 1 = true)

This setting controls whether to allow a SWF file produced for Flash Player 6 and earlier to execute an operation that has been restricted in a newer version of Flash Player.

Flash Player 6 made security sandbox distinctions based on superdomains. For example, SWF files from www.example.com and store.example.com were placed in the same sandbox. Flash Player 7 and later have made security sandbox distinctions based on exact domains, so, for example, a SWF file from www.example.com is placed in a different sandbox than a SWF file from store.example.com. The exact-domain behavior is more secure, but occasionally users may encounter a set of cooperating SWF files that were created when the older superdomain rules were in effect, and require the superdomain rules to work correctly.

When this occurs, by default, Flash Player shows a dialog box asking users whether to allow or deny access between the two domains. Users may configure a permanent answer to this question by selecting Never Ask Again in the dialog, or by visiting the Settings Manager. The LegacyDomainMatching setting lets you override users' decisions about this situation.

This setting does not have a default value. If it is not included in the mms.cfg file, the user can determine whether to allow the operation in a global manner (using the Settings Manager), or on a case-by-case basis (using an interactive dialog box). The values the user can choose among are "Ask," "Allow," and "Deny." The default value is "Ask".

If this value is set to 1, Flash Player behaves as though the user answers "allow" whenever they make this decision. If it is set to 0, Flash Player behaves as though the user answers "deny" whenever they make this decision.

LocalFileLegacyAction

`LocalFileLegacyAction = [ 0, 1 ]` (0=false, 1=true)

This setting controls how Flash Player determines whether to execute certain local SWF files that were originally produced for Flash Player 7 and earlier.

Flash Player 7 and earlier placed all local SWF files in the local-trusted sandbox. Flash Player 8 and later have, by default, placed local SWF files in either the local-with-filesystem or local-with-networking sandbox. In order for a SWF file to be placed in the local-trusted sandbox in Flash Player 8 or later, that SWF file must be designated trusted, using either the Settings Manager or a trust configuration file. This latter behavior is more secure, but occasionally users may encounter an older local SWF file that was created when the older local-trusted behavior was in effect, and must be in the local-trusted sandbox in order to work correctly. Users are notified of such situations by a dialog box, but the dialog is only a failure notification, not a means to trust the SWF file in question.

Users can restore the functionality of such SWF files on a case-by-case basis by designating them trusted in the Settings Manager, but if users encounter a large number of such files, they may also elect in the Settings Manager to place all local SWF files published for Flash Player 7 or earlier into the local-trusted sandbox. The LocalFileLegacyAction setting lets you override users' decisions about this situation.

This setting does not have a default value. If it is not included in the mms.cfg file, the user can use the Settings Manager to specify whether to place all older local SWF files into the local-trusted sandbox.

If this value is set to 1 (the most permissive setting), Flash Player behaves as though users had elected to place all older local SWF files into the local-trusted sandbox. If this value is set to 0 (the most restrictive setting), Flash Player behaves as though users had elected never to automatically place older local SWF files into the local-trusted sandbox, and also suppresses the failure notification dialog.

AllowUserLocalTrust

This setting lets you prevent users from designating any files on local file systems as trusted (that is, placing them into the local-trusted sandbox). This setting applies to SWF files published for any version of Flash.

`AllowUserLocalTrust = [ 0, 1 ]` (0=false, **1=true**)

If this value is set to 1 (the default), Flash Player allows the user to specify whether local files can be placed into the local-trusted sandbox, through the use of the Settings Manager Global Security Settings panel and user trust files. If this value is set to 0, the user cannot place files into the local-trusted sandbox. That is, the Settings Manager Global Security Settings panel and user trust files are ignored.

EnforceLocalSecurityInActiveXHostApp

`EnforceLocalSecurityInActiveXHostApp = "executable filename”`

By default, local security is disabled whenever the ActiveX control is running in a non-browser host application. In rare cases when this causes a problem, you can use this setting to enforce local security rules for the specified application. You can enforce local security for multiple applications by entering a separate `EnforceLocalSecurityInActiveXHostApp` entry for each application.

The filename string must specify the executable filename only, not the full path to the executable; if you specify a full path, the setting is ignored. You can optionally include the EXE (Windows) or APP (Macintosh) file extension. On the Macintosh, you can specify either the name of the actual executable or the name of an application bundle within which the executable is located.

The text encoding of mms.cfg is significant when specified filenames include non-ASCII characters; see “Character encoding” on page 61.

DisableNetworkAndFilesystemInHostApp

`DisableNetworkAndFilesystemInHostApp = "executable filename”`

This option is similar to EnforceLocalSecurityInActiveXHostApp, but applies to plug-ins as well as the ActiveX control, and imposes stricter security controls. When a plug-in or ActiveX control is running within an application specified, it will be as though the HTML parameter `allowNetworking="none"` had been specified. That is, no networking or file system access of any kind will be permitted, and the SWF running in the Flash Player will run without the ability to load any additional media or communicate with any servers. You can enforce local security for multiple applications by entering a separate `DisableNetworkAndFilesystemInHostApp` entry for each application.

The filename string must specify the executable filename only, not the full path to the executable; if you specify a full path, the setting is ignored. You can optionally include the EXE (Windows) or APP (Macintosh) extension. On the Macintosh, you can specify either the name of the actual executable or the name of an application bundle within which the executable is located.

The text encoding of mms.cfg is significant when specified filenames include non-ASCII characters; see “Character encoding” on page 61.

Socket connection options

These settings determine whether socket connections using the ActionScript Socket and XMLSocket classes are permitted. Socket connections also require the presence of a socket policy file on the target server; for more information, see "Data loading through different domains" on page 90.

DisableSockets

`DisableSockets = [ 0, 1 ]` (**0 = false**, 1 = true)

This option enables or disables the use of the `Socket.connect()` and `XMLSocket.connect()` methods. If you don't include this option in the mms.cfg file, or if its value is set to 0, socket connections are permitted to any server. If this value is set to 1, no socket connections are allowed. However, if you want to disable some but not all socket connections, set this value to 1 and then use EnableSocketsTo to specify one or more servers to which socket connections can be made.

EnableSocketsTo

`EnableSocketsTo = [ host name, IP address ]`

This option is effective only when `DisableSockets` has a value of 1; it creates a whitelist of servers to which socket connections are allowed. Unlike most other mms.cfg options, you can use this option as many times as is appropriate for your environment. Note that the servers specified are target servers, to which socket connections are made; they are not origin servers, from which the connecting SWF files are served.

The values specified here must exactly match the values specified in the ActionScript `connect()` methods. If you specify an IP address here, but the `connect()` method specifies a host name, the method fails even if that host name resolves to the specified IP address. Similarly, if you specify a host name here but the `connect()` method specifies an IP address, the method fails.

Using this option does not take the place of a socket policy file on the target server. That is, this option has no effect if the specified server does not have a socket policy file.

GPU Compositing

Flash Player rendering can use the graphics processor unit (GPU) on the video card to accelerate image compositing. In certain circumstances, Flash Player disables GPU compositing. The option in this section lets you override this action and enable GPU compositing.

OverrideGPUValidation

`OverrideGPUValidation = [ 0, 1 ]` (**0 = false**, 1 = true)

The GPU compositing feature is gated by the driver version for video cards. If a card and driver combination does not match the requirements needed to implement compositing, set OverrideGPUValidation to 1 to override validation of the driver requirements. For example, you might want GPU compositing enabled during a specific test suite, even if the video driver in the test machine doesn't meet compositing requirements. This setting overrides driver version gating but still checks for VRAM requirements.

Adobe recommends that you use this setting with care. Overriding GPU validation can result in rendering problems or system crashes due to driver issues. After completing the tests or programming tasks that require the use of this setting, consider setting it back to 0 (or removing it from the mms.cfg file) for normal operations.

RTMFP options

The mms.cfg options described in this section let you specify settings related to peer-to-peer (P2P) connections and the Real Time Media Flow Protocol (RTMFP). For more information about RTMFP, see the FAQ at www.adobe.com/go/rtmfp_faq.

RTMFPP2PDisable

`RTMFPP2PDisable = [ 0, 1 ]` (**0 = false**, 1 = true)

This option specifies how the NetStream constructor connects to a server when a value is specified for *peerID*, the second parameter passed to the constructor. If RTMFPP2PDisable has a value of 0 or is not present in the mms.cfg file, a peer-to-peer (P2P) connection can be used. If this value is 1, any value specified for *peerID* is ignored and P2P connections are disabled; NetStream objects can connect only to Flash Media Server.

RTMFPTURNProxy

`RTMFPTURNProxy = URL of TURN proxy server`

If this option is present, Flash Player attempts to make RTMFP connections through the specified TURN server in addition to normal UDP sockets. TURN Servers are useful for conveying RTMFP network traffic through firewalls that otherwise block UDP packets.

The Global FlashPlayerTrust directory

Application installers can specify that certain files or directories of files that are stored on the user's computer should be *trusted* for all users, and be placed in a local-trusted sandbox. (For a discussion of sandboxes, see "Security sandboxes for local content" on page 87.) If you are deploying applications with content that should be trusted for all users on a computer, you can place trust information for that application in a directory that you specify as a trusted directory. Because information in this directory applies to all users, the directory requires administrative access.

This directory is named FlashPlayerTrust, and is called the Global FlashPlayerTrust directory. It is located alongside the directory that contains the mms.cfg file (see "mms.cfg file location" on page 60). For example, if the mms.cfg file is in C:\Windows\System32\Macromed\Flash, the location of the Global FlashPlayerTrust directory is C:\Windows\System32\Macromed\FlashPlayerTrust. (For information on specifying content as trusted only for the current user, see "The User FlashPlayerTrust directory" on page 82.)

The Global FlashPlayerTrust directory can contain any number of trust configuration files. At startup, Flash Player reads all files in this directory. The names of these files are unimportant; you can choose any filenames you want for your trust configuration files. Generally, each file contains information on a single application, but you can put information on several applications in a single file if you prefer. The configuration file is a text file; each line contains the name of a file or directory, to be trusted. If you specify a directory, all files at or below that directory level are trusted.

To create a configuration file to trust a file or directory:

1. Create a new file in the Global FlashPlayerTrust directory using a text editor, and save it with a unique name.

   Choose a name for your trust configuration file that is unlikely to collide with the names of any other trust configuration files that might be installed. One good way to do this is to name the file after the particular product you are trusting. For example, if you are trusting an employee vacation application, you might call the trust configuration file EmployeeVacation.cfg.

2. Type or paste each directory path (any directory path on the user's hard disk) or file name on a new line in the file. You can paste multiple directory paths on separate lines. When you finish, your file might look similar to the following:

```
# Trust all files in the Employee online calendar app
C:\Program Files\Personnel\Employees\OnlineCalendar
# Trust the file that checks remaining vacation days for an employee
C:\Program Files\Personnel\Employees\VacationDaysRemaining.swf
```

In this example, the SWF file is not in the same directory as the online calendar app, so it must be trusted separately.

3. Save your changes.
4. To test whether the files have been trusted correctly, you can do one of the following:
   - Run the SWF file named in the configuration file.
   - Create a SWF file in the trusted directory that displays the value returned by the ActionScript API `System.security.sandboxType` (ActionScript 1.0 or 2.0) or `Security.sandboxType` (ActionScript 3.0). Run the SWF file in a browser, not through the use of the Test Movie command in Flash. (When SWF files run via Test Movie, local security is not implemented.) The value should be `"localTrusted"`.

CHAPTER 4

User-Configured Settings

4

This chapter provides information on options end users can set for managing privacy and security settings when running Adobe Flash Player on their computers.

This chapter includes the following sections:

Accessing user settings 77
Privacy options 78
Local storage options 79
Update options 80
Security options 80
The User FlashPlayerTrust directory 82

Accessing user settings

Flash Player lets users make a number of decisions regarding privacy, local storage, and so on. These settings are available to the user in three primary ways:

- Pop-up dialogs that appear when Flash Player tries to perform an activity that requires user consent, such as accessing a camera or saving data to disk
- A tabbed set of dialogs that the user can display by right-clicking (command-clicking on the Macintosh) and choosing Settings from the context menu
- The Flash Player Settings Manager, a set of web pages that lets users specify preferences for all available settings.

NOTE	Although the Settings Manager appears within the adobe.com website, it manages only local settings on users' computers, and does not transmit any information back to adobe.com. The Settings Manager is essentially a local control panel that is delivered via the adobe.com website. Flash Player takes great care to ensure that only the official Adobe Settings Manager application is capable of reading or altering users' settings.

In many cases, you can use the mms.cfg file to override user-specified settings, and implement more stringent or more accessible settings. For more information, see Chapter 3, "Administrator Settings."

NOTE	If you use the mms.cfg file to override user settings, the mms.cfg settings are not displayed to the end user. That is, users may think they are specifying a setting when, in fact, their choices are not being honored. If you think this might be confusing for your users, you might want to let them know that certain settings are unavailable to them.

Much of the information in this chapter is excerpted from the online help pages for Flash Player settings. The help pages are geared towards end users, and provide additional explanatory information that might help you or your users more fully understand certain options that are available. The home page for Flash Player help is www.adobe.com/go/player_help_en.

NOTE	In the following sections, screen shots are provided to illustrate the pop-up dialog boxes and the tabbed Settings Panels. For Settings Manager pages, links are provided instead of screen shots, so you can navigate to that page and see the actual Settings Manager online.

Privacy options

Privacy options let the user specify whether an application can have access to the camera or microphone. Users specify these options in one of several ways, summarized below. You can use the AVHardwareDisable option in the mms.cfg file to override user privacy settings.

- The first time a site tries to access the camera or microphone, a pop-up dialog appears. This dialog lets the user specify a one-time preference to allow or deny access.

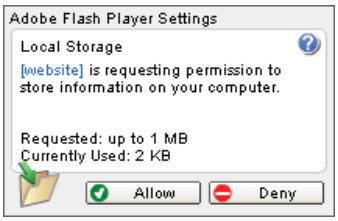
- The Privacy tab lets the user allow or deny access to the camera and microphone for all applications from the current website without asking for permission each time.

- The Website Privacy Settings Panel at www.adobe.com/go/website_privacy_settings lets the user specify settings for any of the web sites that have already requested permission to use the camera or microphone.
- The Global Privacy Settings Panel at www.adobe.com/go/global_privacy_settings lets the user reset privacy options for all web sites.

Local storage options

Local storage options let the user specify whether an application can place a *shared object* on their computer, and the maximum size that object can attain. Applications use shared objects to store data such as user names, game scores, shopping preferences, and so on. (For more information on local shared objects, see the article entitled "What are local shared objects?" at www.adobe.com/products/flashplayer/articles/lso. Users specify these options in one of several ways, summarized below. You can use a number of options in the mms.cfg file to override user local storage settings; see "Data loading and storage options" on page 64.

- The first time a site tries to store information on the user's computer, a pop-up dialog appears. This dialog lets the user specify a one-time preference to allow or deny access.



- The Local Storage tab lets the user allow or deny access for local storage for all applications from the current website without asking for permission each time.

- The Website Storage Settings Panel at www.adobe.com/go/website_storage_settings lets the user specify storage settings for any of the web sites that have already requested permission to store data locally.
- The Global Storage Settings Panel at www.adobe.com/go/global_storage_settings lets the user specify storage settings for any web sites that have not yet requested permission to store data locally. This panel also lets the user choose whether to store data for a third-party local shared objects (objects being stored by a website whose originating domain does not match the URL displayed in the browser window) and whether to store common Flash components to reduce download times.

Update options

Update options let the user specify whether Flash Player should display a notification when a new version is available, and how frequently to check for new versions. You can use the AutoUpdateDisable and AutoUpdateInterval settings in the mms.cfg file to prevent the user from choosing auto-update or to override the frequency of checking for new versions. However, there is no way to ensure that the user doesn't disable auto-update; that is, there is no way to ensure that a user will choose to be notified when a new version of Flash Player is available.

To specify auto-update settings, the user uses the Global Notifications Settings Panel (www.adobe.com/go/global_notification_settings).

Security options

This section describes the security options available to end-users. For more information on Flash Player security in general, see Chapter 5, "Security Considerations." You can use a number of options in the mms.cfg file to override user security options; see "Security options" on page 69.

End users should rarely need to intervene in Flash Player security decisions. However, because the Flash security model evolves over time, occasionally Flash Player encounters a situation in which Flash content attempts to perform an operation that was permitted in a previous version of Flash Player, but is no longer permitted by default. In these situations, it is impossible for Flash Player to tell whether the Flash content in question is legitimate older content that was authored before the change in rules, or malicious content that is attempting to break the newer rules. Flash Player handles these situations conservatively, guiding users toward secure choices, but offering users the ability to restore functionality of older content that has been inadvertently affected.

When Flash content attempts to use older domain matching rules, Flash Player presents a Security dialog box:.

Users may interactively allow or prevent the attempted operation. If they choose "Never ask again", their allow or deny choice is remembered and used for all future instances where this dialog would be presented. Users can later see or change their remembered choice in the Settings Manager at www.adobe.com/go/global_security_settings. Their remembered choice is shown there as "Always ask", "Always allow", or "Always deny".

When Flash content attempts to use older local security rules, Flash Player presents a different dialog box:

This dialog box is only a failure notification - it does not provide an interactive allow option. However, the Settings button in this dialog box brings users to the same Settings Manager link given above. In the Settings Manager, users can affect local security rules in two ways:

- The “Always ask”, “Always allow”, or “Always deny” choice affects not only domain matching, as previously mentioned; it also governs Flash Player's behavior when content attempts to use older local security rules. However, the Ask/Allow/Deny choice affects only content that is apparently older; that is, content that specifies an older version number.
- Users can add local file system paths that are to be placed in the local-trusted sandbox (see “Security sandboxes for local content” on page 87). This enables finer-grained control than the Ask/Allow/Deny choice, and also works for Flash content of any version. Only local paths have any effect in this list; Web domains and URLs have no effect, as remote content may never be placed in a local sandbox. Also, this list, unlike the Ask/Allow/Deny choice, affects only local security rules, not domain matching rules.

Flash Player administrators can use several options in the mms.cfg configuration file to restrict users' ability to make these security choices.

- The LegacyDomainMatching and LocalFileLegacyAction options control Flash Player's behavior in the situations where, respectively, the domain matching or local security dialogs would be displayed. There is only a single user control (Ask/Allow/Deny) for both of these situations, but you can specify different options for each of them using these two mms.cfg options.
- The AllowUserLocalTrust option controls users' ability to add individual paths to the local-trusted sandbox.

For more information on these options, see “Security options” on page 69 in Chapter 3, “Administrator Settings.”

The User FlashPlayerTrust directory

Application installers or end users can specify that certain files or directories of files that are stored on the user’s computer should be *trusted*, and be placed in the user’s local-trusted sandbox. (For a discussion of sandboxes, see “Security sandboxes for local content” on page 87.) Information on these trusted files is stored in a directory called the User FlashPlayerTrust directory. This directory registers files or directories as trusted only for the current user. (For information on registering files as trusted for all users, see “The Global FlashPlayerTrust directory” on page 74.) You can specify whether users can permit applications to be trusted; see “Security options” on page 69.

Information about trusted files can be placed in this directory in two ways:

- An administrator or end-user can create a config file and store it in the User FlashPlayerTrust directory.
- A user without administrative rights can install an application that registers itself as locally trusted.

The User FlashPlayerTrust directory is located in the following location:

- **Windows Vista** C:\Users*username*\AppData\Roaming\Macromedia\Flash Player\#Security\FlashPlayerTrust
- **Windows 2000 and Windows XP** C:\Documents and Settings*username*\Application Data\Macromedia\Flash Player\#Security\FlashPlayerTrust
- **Macintosh** /Users/*username*/Library/Preferences/Macromedia/Flash Player/#Security\FlashPlayerTrust
- **Linux** GNU-Linux ~/.macromedia/#Security/FlashPlayerTrust

For information on how to create and format these configuration files, see "The Global FlashPlayerTrust directory" on page 74.

CHAPTER 5

Security Considerations

5

Clearly, it is critical to maintain the security and integrity of your users' computers when you install Adobe Flash Player. This chapter provides an overview of security, focusing on those aspects of particular interest to administrators deploying Flash Player. Adobe has developed a number of web pages, white papers, chapters in other books, and tech notes that address these security issues, as well as others, in more detail. For a list of these resources, see "Additional security resources" on page 91.

This chapter includes the following sections:

Security overview 85
Security sandboxes for local content 87
About compatibility with previous Flash Player security models 89
Data loading through different domains 90
Additional security resources 91

Security overview

As a computer system administrator, one of your primary responsibilities is to ensure the security and integrity of the data on the systems you manage. Adobe addresses Flash Player security in a number of ways, ranging from settings users can control individually to files that must be placed on servers to allow advanced applications to pass information between different domains.

Because of security issues that arise with relation to Internet access, Adobe (and formerly Macromedia) has implemented more stringent security measures with each release of Flash Player. Through improvements in the security model, Flash Player 10 by default provides much stricter limitations on potentially malicious activities than earlier versions of Flash Player. (In fact, some of these improvements can require you, application authors, or end users to specifically permit actions that were permitted by default in earlier players; see "About compatibility with previous Flash Player security models" on page 89.) Additionally, you can control a number of security-related settings through the use of a config file that you deploy on a user's system when you deploy the player.

Depending on how security settings are permitted or prohibited by the application author, the end user, or you (the administrator), Flash Player may or may not be able to download files to the local disk, upload files from the disk, write shared objects to disk (sometimes referred to as "Flash cookies"), access and run other SWF files on the local disk, or communicate between the local disk and the Internet.

In addition, there are certain activities that Flash Player can never perform, such as reading the path of a local file. For example, even if an application (SWF file) tries to upload or download a file, the application can't set the default file location for the file; the default location shown in the dialog box is the most recently browsed folder, if that location can be determined, or the desktop. Also, the application can't read from or write to the transferred file. In fact, the SWF file that initiated the upload or download can't access the uploaded or downloaded file or even the file's location on the user's disk. Another example is that a SWF file can never determine the contents of a local directory.

With regard to ensuring security of users' computers, the areas of primary interest to administrators are the following:

- How Flash uses security sandboxes to determine whether and how a SWF file on the local disk can communicate with SWF files on the network (see "Security sandboxes for local content" on page 87)
- How users can interactively allow or prohibit certain potentially malicious activities (see Chapter 4, "User-Configured Settings," on page 77)
- How you can deploy a configuration file to override choices users might make with regards to security and privacy issues (see Chapter 3, "Administrator Settings," on page 59)

The area of cross-domain security might also be of interest, although it is usually addressed by application authors. However, authors of applications you plan to deploy might request that you implement a server-side policy file, for example, to permit certain types of cross-domain file access. For more information, see "Data loading through different domains" on page 90.

NOTE

Users who are working in the Flash authoring environment to create applications have access to a number of ways to implement certain security features. These techniques are described in the documentation that accompanies the authoring tool, and are not discussed in this document. If some of your users are developing Flash content, ensure that security measures that you implement are compatible with the features of the applications they are developing, and vice versa.

Security sandboxes for local content

Client computers can obtain individual SWF files from a number of sources, such as by downloading them from external web sites or by copying them from a network server. Flash Player individually assigns local SWF files (those stored on the end-user's computer) and other resources, such as shared objects, bitmaps, sounds, videos, and data files, to *security sandboxes* based on their origin when they are loaded into Flash Player.

Interaction between files in different sandboxes is limited; these limitations prevent SWF files from performing operations that could introduce security breaches. Restricting how a file can interact with the local file system or the network helps keep users' computers and files safe. By default, local SWF files can communicate within the local file system or with the Internet, but not both.

NOTE

The restrictions that are discussed in this section do not affect SWF files that are served from a web site on the Internet.

Local SWF files can have the following levels of permission:

Access the local file system only (default) A local SWF file can read from the local file system and universal naming convention (UNC) network paths but cannot communicate with the Internet. These files are placed into the *local-with-filesystem* sandbox.

Access the network only A Flash author can specify that a SWF file be able to communicate between the local system and the network, but not have access to the local file system where it is installed. These files are placed into the *local-with-networking* sandbox.

Access to the local file system and the network SWF application installers, end users, and administrators can specify that a local SWF file (or multiple SWF files) be able to read from the local file system where it is installed, read and write to and from servers, and cross-script other SWF files on either the network or the local file system. These files are called *trusted*, and are placed into the *local-trusted* sandbox.

Each of these sandboxes is discussed in more detail in the following sections, and in even greater detail in white papers and other documents that are available online; see "Additional security resources" on page 91.

A Flash author can use the API `System.security.sandboxType` (ActionScript 1.0 or 2.0) or `Security.sandboxType` (ActionScript 3.0) to determine the sandbox in which a SWF file is placed. This API must be used while the SWF file is playing in a browser, not through the use of the Test Movie command in Flash. When SWF files run via Test Movie, local security is not implemented.

The local-with-file-system sandbox

By default, Flash Player places all local SWF files, including all legacy local SWF files (earlier than Flash Player 8), in the local-with-file-system sandbox. For some legacy SWF files, operations could be affected by prohibiting outside network access, but this default provides the most secure implementation. (For more information on potential issues with legacy SWF files, see "About compatibility with previous Flash Player security models" on page 89.)

From this sandbox, SWF files may read from files on local file systems or a UNC network path, but they may not communicate with the network in any way. This assures the user that local data cannot be leaked out to the network or otherwise inappropriately shared.

The local-with-networking sandbox

When a Flash author specifies that local SWF files should be assigned to the local-with-networking sandbox, the SWF files are allowed to access the network but forfeit their local file system access. However, a local-with-networking SWF file still is not allowed to read any network-derived data unless permissions are present for that action. That is, a local-with-networking SWF file has no local access, yet it has the ability to transmit data over the network and can read network data from those sites that designate site-specific access permissions.

The local-trusted sandbox

As its name implies, placing files in this sandbox indicates that they can be trusted not to perform any malicious activities that would compromise the security of the local system or of the network. SWF files assigned to the local-trusted sandbox can interact with any other SWF files, and load data from anywhere (remote or local). Files (or entire directories) can be registered as trusted in a number of ways.

- An end-user can respond to a pop-up dialog box or use the Flash Player Settings Manager to specify that a SWF file or set of files should be trusted for that user. For information on settings available to end-users, see "Security options" on page 80. For information on how to control the end-users' ability to specify trusted files, see "Security options" on page 69.
- An administrator, an installer program, or an end-user can create configuration files and place them directly in the appropriate directories. The configuration files are placed in a directory named FlashPlayerTrust on the user's computer, in one of two locations. One location requires administrative access and applies to all users on a computer; see "The Global FlashPlayerTrust directory" on page 74. The other location doesn't require administrative access and applies only to the current user; see "The User FlashPlayerTrust directory" on page 82.

  When an installer installs local SWF files and HTML files, those files should be trusted, because the user consented to run an installer executable to create them. Likewise, when an installer installs an application that plays local SWF files by embedding a Flash Player, the application should be able to play local SWF files in a trusted mode, even if the embedded Flash Player would normally enforce local security. End users should exercise the same caution installing Flash applications as they would when installing any other applications on their computer.

About compatibility with previous Flash Player security models

As a result of the security feature changes over Flash Player's history, content that runs as expected in one Player version might not run as expected in later versions. In these cases, you (and end-users) can specify security settings that are less stringent than the Flash Player default settings. In other words, you can choose to run certain content in a less secure environment.

For example, local SWF files can't communicate with the Internet without a specific configuration on the user's computer. Suppose you have legacy content that was published before these restrictions were in effect. If that content tries to communicate with the network or local file system, or both, Flash Player stops the operation. By default, a Security pop-up question appears, and the user must explicitly provide permission for the application to work properly.

To prevent users from having to provide permission explicitly, Flash provides a number of options.

- An end-user can use the Global Security Settings Panel at www.adobe.com/go/global_security_settings to specify that a file or set of files should be trusted.
- An end-user, or an installer program run without administrative access, can place a local configuration file on the user's machine to specify that a file or set of files should be trusted (see "The User FlashPlayerTrust directory" on page 82).
- You, or an installer program run with administrative access, can place a global configuration file on the user's machine to specify that a file or set of files should be trusted (see "The Global FlashPlayerTrust directory" on page 74).
- You can set an option in a configuration file you deploy to users' machines, the mms.cfg file, to always allow or always deny such access (see "Security options" in Chapter 3, "Administrator Settings.").
- You can run a free, command-line utility called the Local Content Updater on the legacy SWF files. The Local Content Updater lets you change the security sandbox that the SWF file operates in when it is played as a local file in Flash Player 8 and above. It can add, remove, or check for local-with-networking privileges, operating on one or many SWF files. For more information or to download the utility, see Local Content Updater at www.adobe.com/support/flashplayer/downloads.html#lcu.

Data loading through different domains

To make data from a web server available to SWF files from other domains, you may be asked by a Flash author to create a policy file on your server. Policy files are XML files placed in a specific location on your server.

Policy files affect access to a number of assets, including the following:

- Data in bitmaps, sounds, and videos
- Loading XML and text files
- Importing SWF files from other security domains into the security domain of the loading SWF file
- Access to socket and XML socket connections

There are two types of policy files—URL policy files and socket policy files.

- URL policy files provide a way for the server to indicate that its data and documents are available to SWF files served from certain domains or from all domains.
- Socket policy files enable networking directly at the lower TCP socket level, using the Socket and XMLSocket classes.

Requirements for implementing policy files are more strict in Flash Player 10 than in earlier versions of Flash Player. For more information, see the Flash Player Developer Center at www.adobe.com/devnet/flashplayer, as well as the information listed below in "Additional security resources".

Additional security resources

For quick reference, the following list summarizes various web pages and documents related to security, many of which are mentioned elsewhere in this chapter or in other chapters in this book.

- Flash Player Security and Privacy (www.adobe.com/products/flashplayer/security/). This document provides an overview of how Flash Player maintains users' privacy.
- Security Topic Center (www.adobe.com/devnet/security/). This document provides information on security and links to a number of other resources.
- Flash Player Developer Center (www.adobe.com/devnet/flashplayer). This site provides links to a number of security-related documents geared for developers.
- The "Flash Player Security" chapter in *Programming ActionScript 3.0* (www.adobe.com/go/flashcs4_prog_as3_security_en).

- Flash Player 9 Security white paper (www.adobe.com/devnet/flashplayer/articles/flash_player9_security_wp.html). This document focuses on how Flash Player 9.0.124.0 addresses a number of issues related to security, including features previously introduced in earlier versions of the product.

NOTE	At the time of the initial release of Flash Player 10, the security white paper has not been updated for Flash Player 10. For information on security and Flash Player 10, see "Understanding the security changes in Flash Player 10" at www.adobe.com/devnet/flashplayer/articles/fplayer10_security_changes.html.

- Flash Player Help for user setting panels (www.adobe.com/go/player_help_en). These pages explain security settings users can specify using the Settings Manager, settings dialog boxes, and questions that might pop up while a SWF is running.
- The documents titled "What is Flash Player security for local content?" (www.adobe.com/products/flashplayer/articles/localcontent) and "How do I let local Flash content communicate with the Internet?"(www.adobe.com/go/4c093f20). These documents describe the security issues involved in allowing (or preventing) local SWF files from accessing the Internet.
- The Flash Player Local Content Updater (www.adobe.com/support/flashplayer/downloads.html#lcu) lets you change the security sandbox in which SWF files written for Flash Player 7 and earlier operate.
- ActionScript 2.0 and Security (see the "Understanding Security" chapter in *Learning ActionScript 2.0 in Adobe Flash*).