

Adobe® Flash® Player and Adobe AIR® security

Both Adobe Flash Platform runtimes—Flash Player and AIR—include built-in security and privacy features to provide strong protection for your data and privacy, whether you use these Adobe products on your desktop system or mobile device. Adobe constantly advances these protections to incorporate the latest developments in the industry and stay ahead of the continually evolving threat landscape.

To help ensure that potentially malicious activities do not infiltrate your network or its systems, Flash Player includes customizable user and administrative controls. Additionally, developer controls and guidelines for both Flash Player and AIR enable application authors to protect their customers' systems, from the first line of code to the last. For more information on these controls, go to the Flash Platform Developer Center in the Adobe Developer Connection.

Through advanced desktop updating tools, Adobe helps ensure that you have the most current runtime with the latest protections. Plus, our tight relationships and integration with mobile platform vendors, platforms, and update processes help keep your Adobe software current wherever you are and on whatever mobile platform you choose to use.

Adobe Flash Player

Adobe Flash Player versus Adobe AIR

Adobe Flash Player is a cross-platform, browser-based runtime that runs SWF files *within* a web browser on both personal computers and mobile devices.

Adobe AIR is a runtime that enables developers to build standalone applications that run *outside* a web browser on a wide range of devices, including desktops, smartphones, and TVs.

Adobe Flash Player interprets a series of instructions stored in a binary format known as SWF. The SWF files are typically hosted on a server and downloaded via web protocols to be displayed on the client computer using Flash Player when requested. SWF file content can be viewed in a consistent manner across a broad range of platforms, browsers, and devices.

SWF files typically consist of two components: multimedia content and ActionScript® instructions. ActionScript is the programming language of the Adobe Flash Platform. It is also a European Computer Manufacturers Association (ECMA) standard. Originally developed as a way for developers to program interactivity, ActionScript enables efficient programming of Flash Platform applications for everything from simple animations to complex, data-rich, interactive interfaces.

Flash Player also provides a set of underlying capabilities that allows developers to author media and applications that adhere to best practices for security and privacy with respect to the application user, the client computer, the host server, and the respective data.

Adobe AIR

Adobe AIR enables developers to author applications using HTML, JavaScript, or ActionScript in either Adobe Flash Builder® or Flash Professional software. AIR applications run as standalone client applications outside the typical constraints of a web browser and, therefore, have greater access to the device and its operating system, just like other natively installed applications. Adobe AIR applications adhere to a different security context than applications that run in Flash Player in the browser.

Adobe Flash Player security architecture

The Adobe Flash Platform includes safeguards designed to protect all stakeholders—network and system administrators, application end users, website administrators, and application authors—from malicious and unintentional security risks introduced by other stakeholders or third parties.

Security vulnerabilities are typically the result of one of the following:

- Design and implementation bugs, such as buffer overflows and cross-site scripting, which can lead to security holes that are exploitable by malevolent entities or result in unexpected, unwanted application behavior.
- Other stakeholders with access to the system who may attempt to access data for which they do not have rights. By default, Flash Player applications may not view or modify data without explicit permission from the owner of that resource.
- Internet providers that provide services vulnerable to certain classes of attacks, such as Domain Name System (DNS) and packet routing attacks.

Adobe Flash Player sandboxes and domain isolation

The primary security mechanism in the Adobe Flash Player security architecture is the same-origin policy. The same-origin policy used by Flash Player, as well as most common web browsers, logically separates content from different domains into security sandboxes. The sandboxes are designed to ensure that the data you share with one website is not exposed to the other websites that you visit.

In the context of Flash Player, all resources—including file requests, local data storage (shared objects), and any other resources used by a particular domain and its content—are contained within a sandbox. Each sandbox is isolated from the operating system, file system, network, other applications, and even other Flash Player sandbox instances. The stakeholders of each data resource control access to the rights to that resource, ensuring the security of its contents.

Users can request or obtain individual SWF files in a number of ways and from a variety of sources, including downloading them from external websites and copying them from a network server. Flash Player assigns 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 were loaded into Flash Player.

Flash Player limits interaction between files in different sandboxes, thereby preventing SWF files from performing operations that could introduce security breaches. However, the same-origin policy does not cover content on a local device. Therefore, Flash Player provides additional controls for content from the file system or network shares. Restricting how a file can interact with the local file system or the network helps keep users' computers and files secure. By default, local SWF files can communicate within the local file system or with the Internet, but not both.

SWF files may interact with SWF files from other sandboxes—and with servers—but only if explicitly allowed via permission controls set by the resource stakeholders of the other sandbox. For more information, see the *Adobe Flash Player 10 Security white paper*.

Flash Player always enforces the browser's same-origin policy. The same-origin policy is the foundation of browser security that prevents one site from stealing the data you have stored on another site. In areas where Flash Player provides functionality that is not covered by the same-origin policy or existing browser controls, Flash Player requires some form of opt-in by either the end user or data owner. This ensures that user data is protected at all times.

Additional security controls in Adobe Flash Player

Flash Player takes advantage of operating system-specific compiler protections to reduce the risk of malicious individuals writing exploit code to attack your computer. For instance, on Microsoft® Windows®-based systems, Flash Player is compiled to take advantage of Address Space Layout Randomization (ASLR), Data Execution Prevention (DEP), GhostScript (GS), and Structured Exception Handling (SEH). On Linux®-based systems, Flash Player takes advantage of similar features, such as fstack-protector.

Adobe Flash Player also takes advantage of various types of browser-based sandboxes and privilege restrictions:

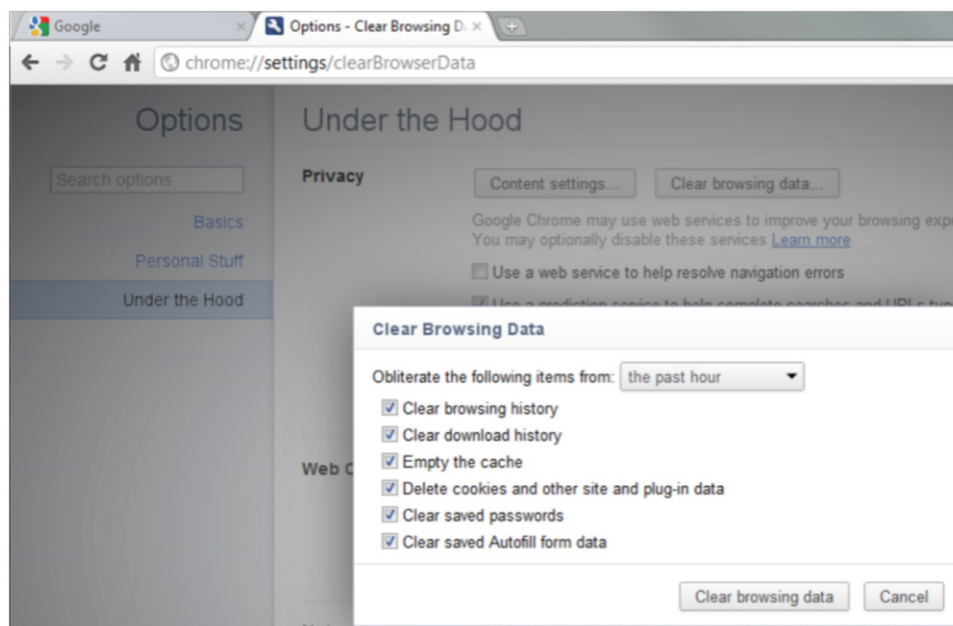
- When executing in Internet Explorer, Flash Player runs in Protected Mode, a mode reserved for low-privilege processes that increases security.
- When executing in Chrome, Flash Player runs within a sandbox that limits it to only the operating system permissions that are necessary to run the application. A broker enforces restrictions on those permissions.
- When executing on a mobile device, such as Android®, Flash Player runs within the browser's application sandbox, which has limited permissions on the device.

By combining compiler flags and sandboxing techniques, Flash Player can take full advantage of protections provided by the operating system to limit the possibility of successful exploits being launched against Flash Player users. To keep up with newly discovered and zero-day threats, Adobe continually advances and expands its sandboxing implementations, increasing its robustness and support in more environments.

Adobe Flash Player privacy protections

To provide an additional layer of privacy protection within the browser environment, Flash Player supports the private browsing mode found in the most popular web browsers, including Internet Explorer, Chrome, Firefox, and Safari. Private browsing lets you browse the web without storing any browsing history on your computer. Flash Player integrates with these web browsers to automatically clear stored data in accordance with the browser's private browsing settings.

In addition, Flash Player integrates with common browser storage deletion options, including the deletion of cookies and local shared objects. For example, users can clear local storage directly from the browser settings interface settings in Internet Explorer, Chrome, Firefox, and Safari.



Adobe Flash Player integrates with browser storage deletion and other security options.

Finally, Adobe Flash Player implements privacy and security by default for camera (webcam) and microphone use. Because these features could cause a privacy violation if automatically enabled, Flash Player always prompts the user for explicit permission to enable the camera or microphone within an application.



Flash Player implements privacy and security for webcams and microphones by default.

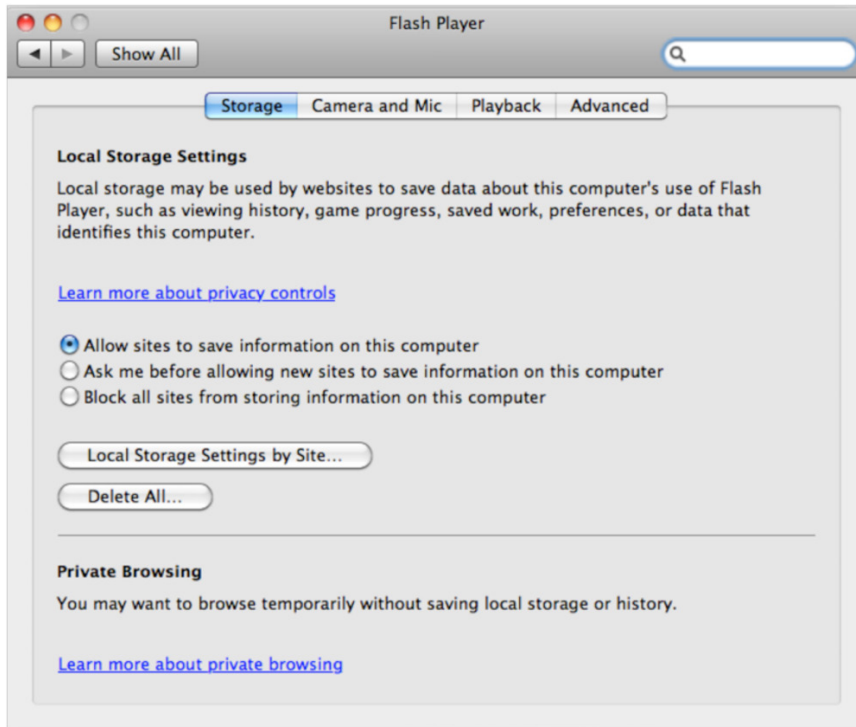
Adobe Flash Player user controls

End users can interactively allow or prohibit certain potentially malicious activities through a range of user-configured settings in the Settings Manager. The Settings Manager allows individual users to specify various security, privacy, and resource usage settings for Adobe Flash technology applications executing on their client computer. For example, users can control application access to specific functions, such as the webcam and microphone, or control the amount of disk space allotted to a SWF file's domain.

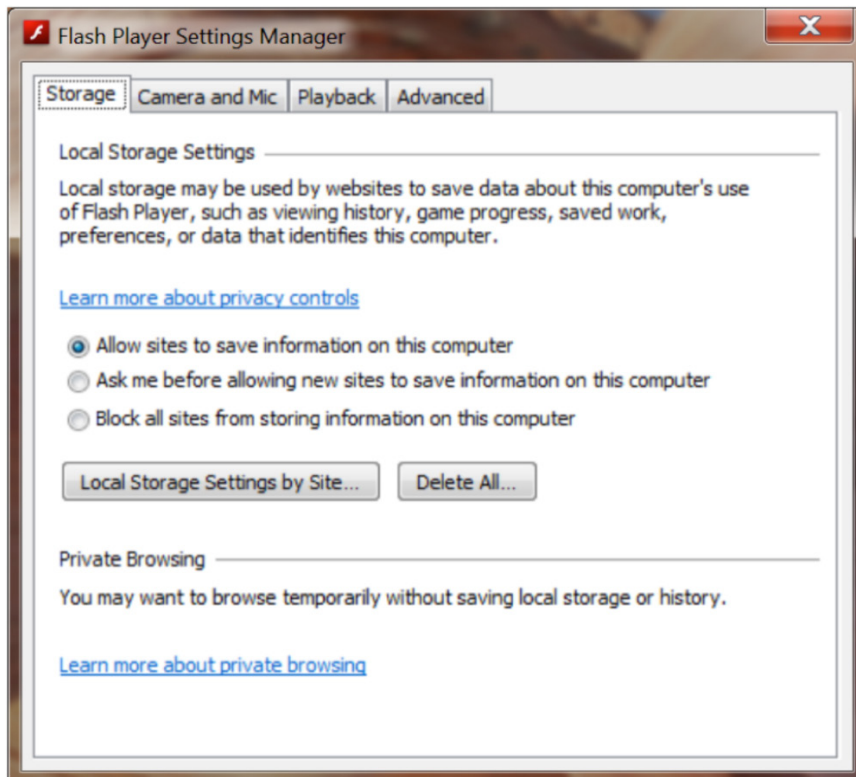
Prior to Flash Player 10.3, the Settings Manager was an online control panel, located on the Adobe website. In Flash Player 10.3 and later, the Settings Manager is a local control panel, enabling end users to manage their Flash Player settings directly on their desktops. With Flash Player 10.3, the Settings Manager is also in the operating system Control Panel or system preferences. Adobe encourages the use of the local Settings Manager.

The local Settings Manager emulates the design paradigm of the client operating system, making it more intuitive to the user. Users can indicate their personal settings either globally (for Flash Player itself and all Adobe Flash technology applications) or specifically (applying to particular domains only). To designate their choices, users select from six categories:

- Global Privacy Settings
- Global Storage Settings
- Global Security Settings
- Global Notifications Settings
- Website Privacy Settings
- Website Storage Settings



The local Settings Manager on Mac OS X.



The local Settings Manager on Windows 7.

For details on managing settings through the local Settings Manager (for Flash Player 10.3 and later), visit http://help.adobe.com/en_US/FlashPlayer/LSM/index.html.

For details on the online Settings Manager (for Flash Player versions prior to 10.3), visit www.adobe.com/support/documentation/en/flashplayer/help/settings_manager.html.

How to access the mms.cfg file

Windows XP, Windows Vista, and Windows 7—The mms.cfg file is located in the Adobe Flash Player folder within the system directory (for example, C:\windows\system32\macromed\flash\mms.cfg on a default Windows XP installation).

Mac OS X—/Library/Application Support/Macromedia/mms.cfg

Linux—The mms.cfg file is flagged as requiring system administrator (or root) permissions to modify or delete the file.

Where to find the Global Flash Player Trust directory

To access the Global Flash Player Trust directory, you must have administrator access to the Adobe Flash Player application.

Windows XP, Windows Vista, and Windows 7—system\Macromed\Flash\FlashPlayerTrust (C:\windows\system32\Macromed\Flash\FlashPlayerTrust on a default Windows XP installation)

Mac OS X—app support/Macromedia/FlashPlayerTrust (for example, /Library/Application Support/Macromedia/FlashPlayerTrust)

Linux—/etc/adobe

Adobe Flash Player administrative controls

System and network administrators can control Flash Player options and Adobe Flash technology applications access via two methods: the mms.cfg file and Global Flash Player Trust directory files. Using these files, administrators can override choices made by end users in the Settings Manager that might introduce security or privacy issues.

mms.cfg file

The mms.cfg file is a system configuration file that enables IT departments within enterprise environments to install Flash Player across the enterprise while enforcing global security and privacy settings.

When Flash Player starts, it reads the security settings from this file and uses them to limit or change functionality. The mms.cfg file includes several types of controls, such as data loading, privacy, updating, and socket controls.

Some mms.cfg options might override settings specified by end users in their Settings Manager, including display, privacy, local storage, microphone, and camera settings. This protects the enterprise against users unwittingly providing access to malicious or dangerous executable files through their settings.

For details on mms.cfg file settings, visit www.adobe.com/go/flash_player_admin.

The Global Flash Player Trust directory

Local files are placed in the local-trusted sandbox only at the direction of a user, an administrative user, or an installer program. Administrative users, and installer programs that run with administrative rights, have the option of designating trust by creating configuration files in the Global Flash Player Trust directory.

Adobe Flash Player reads all configuration files in this directory. Although you can choose any name for the configuration files, to avoid naming conflicts or collisions, the recommended convention is to use the .cfg extension, along with a filename that describes the files being trusted.

Each line of each configuration file lists a local, trusted path. These paths can name individual files or entire directories. If the path specifies a directory, the directory and all of its descendant directories are trusted. Configuration files in the Global Flash Player Trust directory affect all users of the system or computer.

Adobe AIR security architecture

In contrast to Flash Player, which runs within a browser and inherits the browser's security capabilities like other web applications, the primary security model for Adobe AIR is that of a desktop application. Desktop applications inherently have more privileges than a web application, because they have been installed by the user on a specific system, thereby implying a degree of trust that is greater than that of arbitrary web content. On the other hand, the privileges inherent in a desktop application require a higher degree of caution because certain coding practices and patterns that might be common in web applications are not acceptable in desktop applications.

Code signing

To ensure their authenticity, all AIR applications must be signed by a code-signing certificate. This certificate can either be a self-signed certificate, which is not automatically recognized as trustworthy by typical machine (unless users import that specific certificate into their certificate trust store), or a commercial code-signing certificate purchased from a major certificate authority (CA), such as Thawte, VeriSign, GlobalSign, or ChosenSecurity.

Adobe recommends that AIR application developers use a commercial code-signing certificate, which enables the AIR runtime to display the company name associated with that certificate, giving end users peace of mind that they have not downloaded a rogue application attempting to impersonate a valid product.

In contrast, if an AIR application developer uses a self-signed certificate, no information regarding the source of the application is provided to the end user, and a stronger warning message is displayed, informing the user that the AIR runtime cannot verify the application author.

Adobe AIR sandboxes

AIR applications can be built using a combination of Flash technology and HTML or Ajax. In addition, AIR applications can leverage PDF for document rendering, although an AIR application cannot be based on a PDF file alone.

Regardless of whether the application is built primarily in Flash technology or HTML, all AIR applications include a set of AIR privileged APIs that are available to provide access to local system and network resources that would not be normally available in a web application contained in a browser. Each AIR application also contains two different sandboxes, depending on what type of content is being loaded and for what purpose:

- **Application sandbox**—Permits access to the privileged AIR system APIs. In return for access to these powerful APIs, some common dangerous APIs and patterns are restricted. For example, dynamic importing of remote content is generally prohibited, and dynamic code generation techniques are heavily restricted. Only content loaded directly from the application home directory (via the `app:/` URI scheme), that is, trusted content from the signed application, can be placed in the application sandbox.
- **Non-application sandbox**—Contains all other content that is not loaded directly into the application sandbox, including local and remote content. This content does not have direct access to the privileged AIR APIs and must obey the same rules that it would have to obey in the browser when loaded from the same location (for example, a local SWF file behaves the same way a local SWF file would in the browser, and HTML from a remote domain behaves like it would behave in the browser). Although the application author can choose to manually override these default settings, best practices are to start from the position of ultimate security.

The Adobe AIR sandboxes are similar in concept to the same-origin policy sandboxes described for Flash Player. Because the privileged APIs have access to local data from the user's machine, AIR uses sandboxes to ensure that the local data is not accidentally made available to remote content. By placing the privileged APIs in the trusted application sandbox and limiting the types of activities that can occur within that sandbox, AIR makes it easier for developers to protect end-user information.

Integration with mobile platform security mechanisms

Both the Flash Player and AIR runtimes integrate with the built-in security mechanisms implemented on common mobile browsers and operating systems.

While Flash Player inherits the typical permissions of the web browser on mobile platforms, AIR applications are restricted to the functionality allowed at the time of installation. This protects users against individuals attempting to alter a signed AIR application and download malicious code to the user's mobile device via an AIR application.

Updating Flash Player and AIR

The majority of attacks are exploiting software installations that are not up to date on the latest security updates. Adobe therefore strongly recommends that users install the latest security updates as the best possible defense against those with malicious intent.

On the desktop

Both the Flash Player and AIR runtime applications include automatic update notification mechanisms. By default, both applications automatically check for new updates every seven days.

On mobile platforms

Because the application delivery model for each mobile platform on the market is different, how you update Flash Player and AIR varies as well.

For mobile platforms that support updates through their marketplaces, such as Android, Flash Player and AIR updates are available through the marketplace. Adobe provides the updated runtime application to the marketplace, and users are responsible for downloading the new version to their device. Adobe recommends that end users allow automatic updating of Flash Player and AIR through the market application on the Android device.

For mobile platforms on which the AIR runtime is bundled with the application, such as applications available for iOS, the application developer is responsible for updating AIR by recompiling the runtime application with the updated version of the software development kit and delivering the update to the end user.

On platforms on which the Flash Player and AIR runtimes are bundled with the operating system, such as on BlackBerry® Playbook, Adobe provides the updated runtime to the operating system provider, who then recompiles the Adobe runtime into a new version of its operating system. The provider is then responsible for pushing the new operating system version to its customers.

Summary

To stay ahead of the continually evolving threat landscape, Adobe constantly advances the built-in security and privacy features in Adobe Flash Player and Adobe AIR by incorporating the latest developments in the industry. Advanced desktop updating tools help ensure that you have the most current runtime with the latest protections.

Whether you use these Adobe products on your desktop system or a mobile device, both Flash Player and AIR provide strong protection for your data and your privacy.

Customizable user and administrative controls in Flash Player ensure that potentially malicious programs or activities do not infiltrate your network or systems. Plus, our tight relationships and integration with mobile platform vendors, platforms, and update processes help keep your Adobe software current wherever you are and on whatever mobile platform you choose to use.



Adobe

Adobe Systems Incorporated
345 Park Avenue
San Jose, CA 95110-2704
USA
www.adobe.com

Adobe, the Adobe logo, ActionScript, Adobe AIR, AIR, Flash, and Flash Builder 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 U.S. and other countries. Android is a trademark of Google Inc. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Microsoft, Windows, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The Trademark BlackBerry is owned by Research In Motion Limited and is registered in the United States and may be pending or registered in other countries. All other trademarks are the property of their respective owners.

91064059 11/11