BETA CONTENT: Planning and Deploying Windows AppLocker Policies

Microsoft Corporation

Status: Preliminary documentation

Beta content: This guide is currently in beta form. The AppLocker team greatly appreciates you reviewing the document and looks forward to receiving your feedback. To report bugs or ask questions about any of the content in this guide, please send e-mail to the Applocker Feedback alias applock@microsoft.com.

Abstract: This planning and deployment guide provides guidance for completing a functional Windows® AppLocker™ deployment in Windows 7 and Windows Server 2008 R2.

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

Contents

Planning and Deploying Windows AppLocker Policies

Beta content: This guide is currently in beta form. The AppLocker team greatly appreciates you reviewing the document and looks forward to receiving your feedback. To report bugs or ask questions about any of the content in this guide, please send e-mail to the Applocker Feedback alias applock@microsoft.com.

AppLocker in Microsoft Windows Server 2008 R2, Windows 7 Ultimate, and Windows 7 Enterprise helps administrators control which applications are allowed to run in their organization. It does this by allowing administrators to create an approved list of applications in the form of AppLocker rules to control what executable, Windows Installer, script, and DLL files a user can run. Each of these files types is separated into four rule collections, and the rule sets defined in these collections are enforced independently.

Who Should Use AppLocker?

AppLocker is a good solution for organizations that need to perform any of the following tasks:

- Control what applications are allowed to run within the organization
- Control which users are allowed to run licensed applications
- Provide an audit log of what applications users have been running
- Prevent standard users from installing per-user software

When AppLocker is enabled, Windows first checks the AppLocker rule list to determine whether the file is permitted to run. Administrators must carefully plan their AppLocker rules deployment so that specified users can run applications required for their duties and be excluded from running applications not necessary for their duties.

In many organizations, information is the most valuable asset, and ensuring that only approved people have access to that information is imperative. Access control technologies like Rights Management Services (RMS) and access control lists (ACLs) help control what users are allowed to access; however, when a user runs a process, that process is given the same level of access to data as the user. As a result, sensitive information can easily be deleted or transmitted out of an organization simply because the user unknowingly ran malicious software. Administrators can configure AppLocker to help mitigate this problem by only allowing users to run known applications. For more information about these types of attacks, see the Microsoft Security Intelligence Report (http://download.microsoft.com/download/b/2/9/b29bee13-ceca-48f0-b4ad-53cf85f325e8/Microsoft_Security_Intelligence_Report_v5.pdf).

In addition, software publishers are beginning to create more applications that can be installed by standard users (non-administrators). In fact, this type of software deployment often violates an organization's written security policy and circumvents traditional application deployment solutions that only allow software to be installed in controlled locations. By allowing administrators to create a list of application types that are allowed to run, AppLocker rules can prevent such per-user applications from running.

For more information about how AppLocker works and how to set up AppLocker in a test lab, see the following resources:

- What's New in AppLocker in Windows Server 2008 R2 and Windows 7 (http://technet.microsoft.com/en-us/library/dd378941.aspx)
- AppLocker Step-by-Step Guide (http://technet.microsoft.com/en-us/library/dd723686(WS.10).aspx)

About this guide

This guide provides recommendations and steps to help you plan and implement a deployment of AppLocker rules, based on the requirements of your organization, by highlighting your main decision points. Because AppLocker contains a dynamic set of rules for applications, this guide shows you how you can use reference computers to create rules for installed applications. Before you read this guide, you should have a good understanding of how AppLocker works on a functional level. You should also have a good understanding of the organization requirements that will be reflected in your AppLocker design.

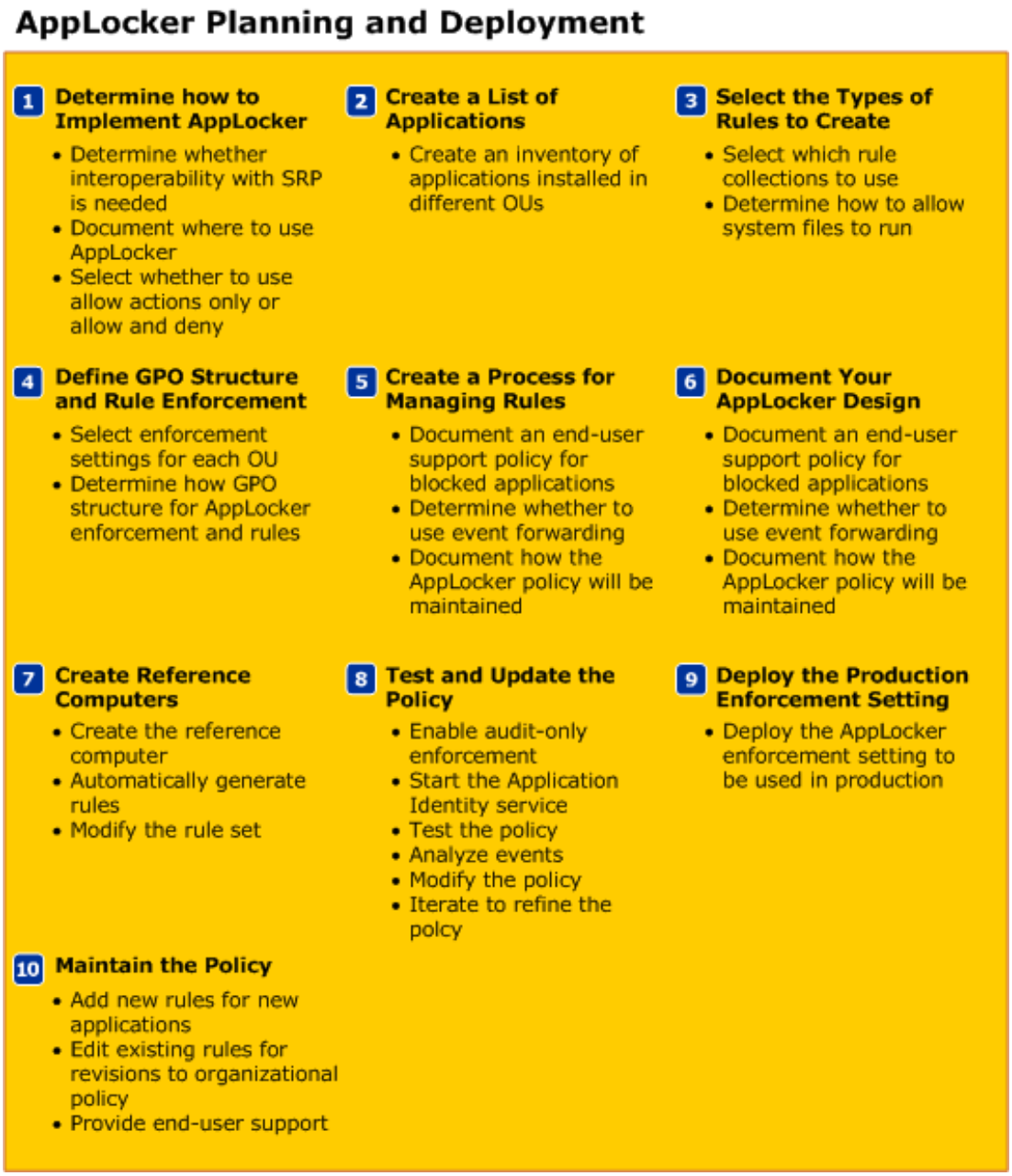
Filing bugs for this guide

This guide is currently in beta form. The AppLocker team greatly appreciates you reviewing the document and looks forward to receiving your feedback. To report bugs or ask questions about any of the content in this guide, please send e-mail to applock@microsoft.com.

Windows AppLocker Policies Planning and Deployment Process

Designing an AppLocker deployment begins by performing several important tasks:

1. Determine how to implement AppLocker
2. Create a list of applications
3. Determine the collections and conditions of rules to create
4. Determine GPO structure and rule enforcement
5. Create a process for managing AppLocker policies and rules
6. Document your AppLocker design
7. Create reference computers and automatically generate rules
8. Test and update the policy
9. Deploy the production enforcement setting
10. Maintain the policy



Step 1: Determine How to Implement AppLocker

Before you create an AppLocker design, you should carefully consider where you will deploy AppLocker in your organization.

AppLocker and Software Restriction Policies

Software Restriction Policies (SRP) was introduced in Windows XP to enable administrators to control which files are permitted to run on a computer. Although SRP and AppLocker have the same goal, AppLocker is a major revision of SRP. Therefore, you cannot use AppLocker to manage SRP settings. In addition, if AppLocker and SRP settings are configured in the same GPO, only the AppLocker settings will

be enforced. Therefore, if you have computers running previous versions of Windows operating systems, you will need to use both SRP settings and AppLocker rules in your organization.

> **Note**: You can still create SRP rules for Windows 7 computers. However, if AppLocker rules are defined in the same GPO, then Windows 7 Ultimate, Windows 7 Enterprise, and Windows Server 2008 R2 computers will use the AppLocker rules, and previous versions of Windows will use the SRP settings.

If you have existing SRP rules, you must create new AppLocker rules to migrate that enforcement. For guidance and procedures for performing a migration from SRP to AppLocker, see Migrating from Software Restriction Policies to AppLocker (Document not yet published).

Areas where AppLocker is most effective

AppLocker can help reduce the total cost of ownership for business groups that typically use a finite set of applications, such as human resources and finance departments. At the same time, these departments access highly sensitive information, much of which contains confidential and proprietary information. By using AppLocker to create rules for specific applications that are allowed to run, you can help limit unauthorized applications from accessing this information.

The following characteristics are typical of good candidates on which to enforce AppLocker rules:

- Users run without administrative rights
- Applications are installed using an installation deployment technology

Areas where AppLocker is less effective

Enforcing AppLocker rules is not suited for business groups that must be able to install applications as needed and without approval from the IT department. If one or more organizational unit (OU) in your organization has this requirement, you can either choose not to enforce application rules in those OUs by using AppLocker or to implement an audit only enforcement setting through AppLocker.

> **Note**: AppLocker can also be effective in helping create standardized desktops in organizations where users run as administrators. However, it is important to note that users with administrator privileges can add new rules to the local AppLocker policy.

The following characteristics are typical of areas where enforcing AppLocker rules might not be ideal:

- Users must be able to install applications as needed
- Users currently have administrator access and it would be difficult to change this

Enforcing AppLocker rules in support organizations

Because AppLocker rules can be applied to specific users and groups, you can create rules that block standard users from accessing administrative applications while still allowing users with administrator privileges the flexibility to perform privileged tasks. For instance, you can create a rule for the Everyone group that allows all files within the Windows directory to run while excluding the Windows Registry Editor (regedit.exe). By creating a second rule to allow Registry Editor for the Helpdesk group, your

product support group will be able to run Registry Editor, while other users (who are not a member of the Helpdesk group) cannot.

Allow rules versus deny rules

Unlike SRP, each AppLocker rule collection functions as an allowed list of files. Only the files that are listed within the rule collection will be allowed to run. This configuration makes it easier for administrators to determine what will occur when an AppLocker rule is applied.

You can also create rules that use the deny action. When applying rules, AppLocker first checks whether any explicit deny actions are specified in the rule list. If you have denied a file from running in a rule collection, the deny action will take precedence over any allow action, regardless of where the rule was originally applied (i.e., in what GPO). Because AppLocker functions as an allowed-list by default, if no rule explicitly allows or denies a file from running, AppLocker's default deny action will block the file.

Deny rule considerations

Although you can use AppLocker to create a rule to allow all files to run and then use rules to deny specific files, this configuration is not recommended. The deny action is generally less secure than the allow action since a malicious user could modify the file to invalidate the rule. The following list details security concerns for different rule conditions with deny actions.

- **Publisher rules**: A user could modify the properties of a file (e.g., resign the file with a different certificate).
- **File hash rules**: A user could modify the hash for a file.
- **Path rules**: A user could move the denied file to a different location and run it there.

Important: If you choose to use deny rules, you must ensure that you first create rules that allow the Windows system files to run. AppLocker enforces rules for allowed applications by default, so after one or more rules have been created for a rule collection (impacting the Windows system files), only the applications that are listed as being allowed will be permitted to run. Therefore, creating a single rule in a rule collection to deny a malicious file from running will also deny everything else on the computer from running.

Record findings

After you have analyzed the various groups within your organization, record your findings and decisions about where to deploy AppLocker rules. The following table shows sample information for two business groups within Woodgrove National Bank.

Business group	Bank Tellers
Associated OUs	Teller-East / Teller-West
Implement AppLocker?	Yes
Business group	Human Resources
Associated OUs	HR
Implement AppLocker?	Yes

Step 2: Create a List of Applications

After you have determined to which OUs AppLocker rules will be applied, identify which applications are installed in those OUs. Ensure that you know all of the locations where the application has installed files. For example, Microsoft Office 2007 installs files by default to %ProgramFiles%\Microsoft Office\, which maps to \Program Files\Microsoft Office\.

Record findings

For the groups where you have chosen to implement AppLocker, modify the table that you created earlier to record your findings about installed and required applications.

Business group	Bank Tellers
Associated OUs	Teller-East / Teller-West
Implement AppLocker?	Yes
Applications and install paths	• Teller software – C:\program files\Woodgrove\Teller.exe • Windows files
Business group	Human Resources
Associated OUs	HR-All
Implement AppLocker?	Yes
Applications and install paths	• Time Sheet Organizer – C:\program files\Woodgrove\HR\timesheet.exe • Check Payout – C:\program files\Woodgrove\HR\checkcut.exe • Internet Explorer 7 – C:\program files\Internet Explorer\ • Windows files

Step 3: Determine the Collections and Conditions of Rules to Create

When determining what types of rules that you will create for your groups of users, you should also determine what enforcement setting will be used for each group. Depending upon the way that applications are deployed in a specific group, creating rules for different rule collections will be more applicable.

AppLocker rule collections

When planning your AppLocker rules deployment, you should determine the rule collections where you will be listing the AppLocker rules. AppLocker enforces rules by grouping enforcement for different types of files. AppLocker includes four rule collections:

- **Executable**: .exe and .com
- **Windows Installer**: .msi and .msp
- **Script**: PowerShell (.ps1), Batch files (.bat), command scripts (.cmd), VBScript (.vbs), and JavaScript™ (.js)

- **DLL**: .dll and .ocx

Important: Each application can load several DLLs and AppLocker must check each DLL before it is allowed to run. Therefore, creating DLL rules might cause performance problems on some computers. Denying some DLLs from running can also create application compatibility problems. As a result, the DLL rule collection is not enabled by default.

To enable the DLL rule collection:

1. Open the AppLocker snap-in, expand **Application Control Policies**, and then expand **AppLocker**.
2. Right-click **AppLocker**, and then click **Properties**.
3. Click the **Advanced** tab, and then select the Enable DLL rule collection check box.
4. Click **OK** to apply the changes. The DLL rule collection will be displayed in the console tree.

Choosing AppLocker condition types

There are three types of AppLocker conditions that can be created for files: publisher, path, and file hash. The following table details the strengths of each rule condition.

Rule Condition	Requirements	Considerations
Publisher	File must be digitally signed by a software publisher	Rules that are specified to the version level might have to be updated when a new version of the file is released.
Path	None	Because path rules align to locations within the file system, you should ensure that there are no subdirectories that are writable by non-administrators. For example, if you create a path rule for C:\ with the allow action, any file under that location will be allowed to run, including users' profiles.
File hash	None	Rule must be updated each time that a new version of the file is released.

The following topics detail the items that you should review when selecting which rule conditions you will prefer in your organization.

Is the file digitally signed by a software publisher?

If the file is signed by a software publisher, we recommend that you create rules with publisher conditions. You may still create file hash and path conditions for signed files. However, if the file is not digitally signed by a software publisher, you have three options:

- Sign the file using an internal certificate
- Create a rule using a file hash condition
- Create a rule using a path condition

Note: To determine how many applications are digitally signed on a reference computer, you can use the Get-AppLockerFileInformation PowerShell cmdlet for a directory of files to determine their properties. For example, `Get-AppLockerFileInformation -Directory C:\Windows\ -FileType EXE -recurse` will display the properties for all .exe and .com files under the Windows directory.

What type of condition does your organization prefer?

If your organization is already using SRP to restrict what files users can run, rules using file hash or path conditions are probably already in place.

Publisher

Publisher conditions can only be made for files that are digitally signed. Publisher conditions are easier to maintain than file hash conditions and are generally more secure than path conditions.

- **Advantages**:
  - Does not require frequent updating
  - Can be applied to different values within a certificate
  - A single rule can be used to allow an entire product suite
- **Disadvantages**: File must be signed.

Note: You can use the asterisk (*) wildcard within a publisher rule to specify that any value should be matched. For example, specifying * for the file version will allow the rule to be matched to any file version from the matching file name, product name, and publisher.

File hash

File hash rules use a unique hash that Windows calculates for a file. For files that are not digitally signed, file hash rules are more secure than path rules.

- **Advantages**: Because each file has a unique hash, file hash condition will align with only one file.
- **Disadvantages**: Each time that the file is updated (security patch or upgrade), the file's hash will change. As a result, you must manually update file hash rules.

Path

When creating a rule that uses a deny action, path conditions are less secure for preventing access to a file since a user could easily copy the file to a different location than that specified in the rule.

- **Advantages**: Can easily control many directories or a single file.
- **Disadvantages**: Might be less secure if a rule configured to use a directory path contains subdirectories that are writable by non-administrators.

There are other items to consider when creating path rules:

- AppLocker does not enforce rules that specify paths with short names. You should always specify the full path to a file or folder when creating path rules so that the rule will be properly enforced.

- The * wildcard character can be used within path rules to specify that all matches should be considered when the rule is executed. For example, %ProgramFiles%\Internet Explorer* indicates that all files in and below the Internet Explorer folder will be affected by the rule.
- AppLocker uses path variables for well-known directories in Windows. While two of these path variables use the same format as Windows environment variables, it is important to note that they are not environment variables. The AppLocker engine can only interpret AppLocker path variables. The following table details those path variables.

Windows Directory	AppLocker Path Variable	Matching Environment Variable
Windows	%WINDIR%	%SystemRoot%
System32	%SYSTEM32%	%SystemDirectory%
Windows installation directory	%OSDRIVE%	%SystemDrive%
Program Files	%PROGRAMFILES%	%ProgramFiles% and %ProgramFiles(x86)%
Removable media (e.g., CD, DVD)	%REMOVABLE%	
Hot plug media (e.g., USB flash drive)	%HOT%	

Using the default rules

AppLocker includes default rules for each rule collection. These rules are intended to help ensure that the files that are required for Windows to operate properly are allowed in an AppLocker rule collection.

Important: We recommend that you use the default rules as a template when creating your own rules to allow the Windows directories to run. However, these rules are only meant to function as a starter policy when you are first testing AppLocker rules.

If you require additional application security, you might need to modify the built-in default rule set for your requirements. For example, the default rule to allow all users to run .exe files in the Windows folder is based on a path condition that allows everything under the Windows folder. The Windows directory contains a Temp folder to which the Users group is given the following permissions:

- Traverse folder / execute file
- Create files / write data
- Create folders / append data

These permissions settings are applied to this folder for application compatibility. However, because any user can create files in this location, allowing applications to be run from this location might conflict with your organization's written security policy.

Executable Rules

The following table lists the default rules that are available for the Executable Rules collection.

Purpose	Name	User	Rule Condition Type
Allow members of the local Administrators group access to run all executable files	(Default Rule) All files	BUILTIN\Administrators	Path: *
Allow all users to run executable files in the Windows folder	(Default Rule) All files located in the Windows folder	Everyone	Path: %windir%*
Allow all users to run executable files in the Program Files folder	(Default Rule) All files located in the Program Files folder	Everyone	Path: %programfiles%*

Because all of the default rules for the Executable Rules collection are based on folder paths, all files under those paths will be allowed.

Windows Installer Rules

The following table lists the default rules that are available for the Windows Installer Rules collection.

Purpose	Name	User	Rule Condition Type
Allow members of the local Administrators group to run all Windows Installer files	(Default Rule) All Windows Installer Files	BUILTIN\Administrators	Path: *
Allow all users to run Windows Installer files that are digitally signed	(Default Rule) All digitally signed Windows Installer files	Everyone	Publisher: * (all signed files)
Allow all users to run Windows Installer files that are located in the Windows Installer folder	(Default Rule) All Windows Installer files in %systemdrive%\Windows\Installer	Everyone	Path: %windir%\Installer*

Script Rules

The following table lists the default rules that are available for the Script Rules collection.

Purpose	Name	User	Rule Condition Type
Allows members of the local Administrators group to run all scripts	(Default Rule) All scripts	BUILTIN\Administrators	Path: *
Allow all users to run executable files in the Windows folder	(Default Rule) All files located in the Windows folder	Everyone	Path: %windir%*
Allow all users to run executable files in the Program Files folder	(Default Rule) All files located in the Program Files folder	Everyone	Path: %programfiles%*

DLL Rules

The following table lists the default rules that are available for the DLL Rules collection.

Purpose	Name	User	Rule Condition Type
Allows members of the local Administrators group to run all DLLs	(Default Rule) All DLLs	BUILTIN\Administrators	Path: *
Allow all users to run DLLs files in the Windows folder	(Default Rule) Microsoft Windows DLLs	Everyone	Path: %windir%*
Allow all users to run DLLs in the Program Files folder	(Default Rule) All files located in the Program Files folder	Everyone	Path: %programfiles%*

Record findings

To record your findings for this section, document the following items for each OU:

- Whether your organization will use the built-in default AppLocker rules to allow system files to run
- The types of rule conditions that you will use to create rules, stated in order of preference

The following table details sample data for documenting rule type and rule condition findings.

Business group	Bank Tellers
Associated OUs	Teller-East / Teller-West
Implement AppLocker?	Yes

Applications and install paths	• Teller software – C:\program files\Woodgrove\Teller.exe
	  ○ File is signed; create publisher condition
	• Windows files
	  ○ Create a path exception to default rule to exclude \Windows\Temp

Business group	Human Resources
Associated OUs	HR-All
Implement AppLocker?	Yes

Applications and install paths	• Time Sheet Organizer – C:\program files\Woodgrove\HR\timesheet.exe
	  ○ File is not signed; create file hash condition
	• Check Payout – C:\program files\Woodgrove\HR\checkcut.exe
	  ○ File is signed; create publisher condition
	• Internet Explorer 7 – C:\program files\Internet Explorer\
	  ○ File is signed; create publisher condition
	• Windows files

	○ Use default rule for the Windows path

Step 4: Determine GPO Structure and Rule Enforcement

There are two concepts that you must review in order to properly plan how AppLocker rules should be structured for different OUs:

- Enforcement settings
- Rules and enforcement setting inheritance in Group Policy

When determining how many GPOs to create for applying AppLocker rules in your organization, you should consider how AppLocker creates an effective policy through rule enforcement.

Enforcement settings

Rule enforcement can be configured for each rule collection. By default, if enforcement is not configured and rules are present in a rule collection, those rules will be enforced. The following table details the three AppLocker rule enforcement settings in Group Policy.

Enforcement Setting	Description
Not configured	By default, enforcement is not configured in a rule collection. If rules are present in the corresponding rule collection, they will be enforced. If rule enforcement is configured in a higher-level linked GPO, that enforcement value will override the not configured value.
Enforce rules	Rules are enforced for the rule collection and all events are audited.
Audit only	Rules are only audited. This value is useful when planning and testing AppLocker rules.

Rules and enforcement setting inheritance in Group Policy

Group Policy merges AppLocker policy in two ways:

- **Rules**: Merging is additive. Group Policy does not overwrite or replace rules that are already present in a linked GPO. For example, if the current GPO has 12 rules and a linked GPO has 50 rules, 62 rules will be applied to all computers that receive the AppLocker policy. The default AppLocker rules are merged if they are present in multiple GPOs.

  Important: When determining whether a file is permitted to run, AppLocker processes rules in the following order:

  - **Explicit deny**: An administrator has created a rule to deny a file.
  - **Explicit allow**: An administrator has created a rule to allow a file.
  - **Implicit deny**: This is also called the default deny since all files that are not affected by an allow rule are automatically blocked.

- **Enforcement settings**: The last addition to the policy is applied. For example, if a higher level GPO has the enforcement setting configured to **Enforce rules**, and the closest GPO has the setting configured to **Audit only**, **Audit only** will be enforced. If enforcement is not configured

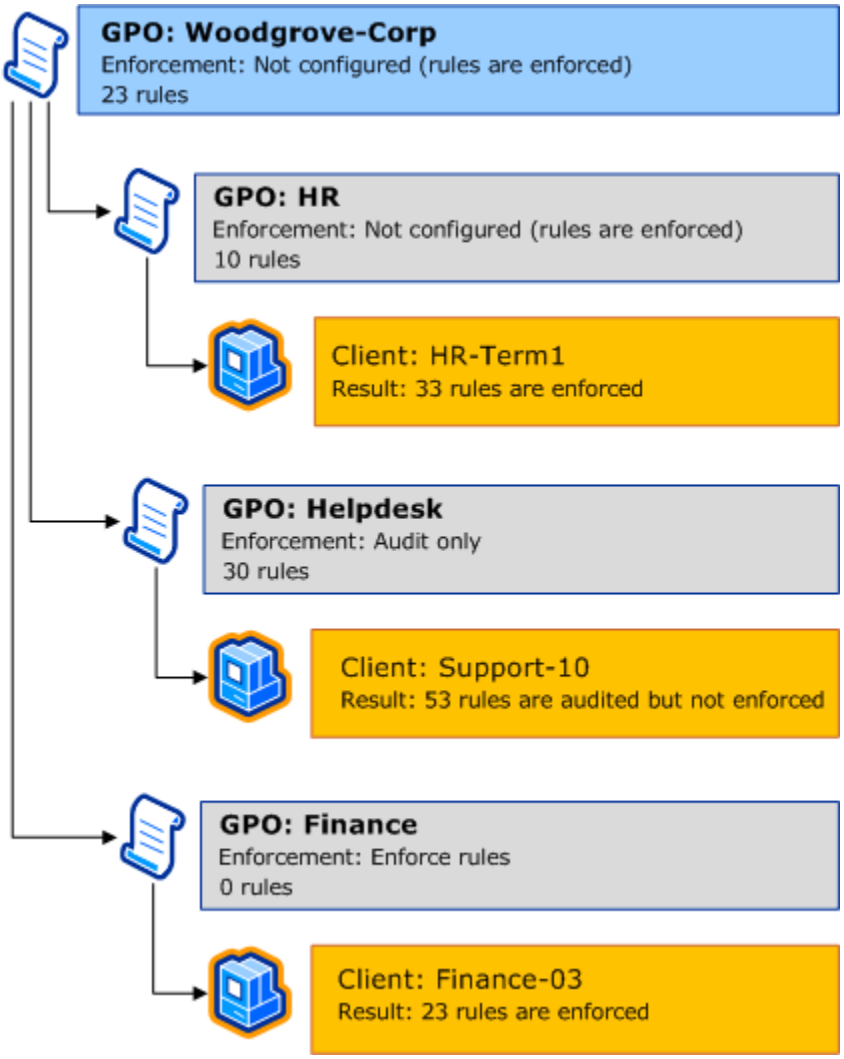
on the closest GPO, the setting from the closest linked GPO will be enforced. AppLocker uses standard Group Policy merging logic, with the settings processed in the following order:

- Local GPO
- Site, including any linked GPOs
- Domain, including any linked GPOs
- OUs

For more information about Group Policy inheritance, see Group Policy Processing and Precedence (http://technet.microsoft.com/en-us/library/cc785665.aspx).

Because a computer's effective policy includes rules from each linked GPO, duplicate rules or conflicting rules could be enforced on a user's computer. Therefore, you should carefully plan your deployment to ensure that only rules that are necessary are present in a GPO.

The following image shows how AppLocker rule enforcement is applied through linked GPOs.



Record findings

After determining how GPOs will be structured for applying AppLocker policy, you should record your findings. You can use the following table to determine how many GPOs to create (or edit) and to which objects they will be linked. When filling in the "Default rules defined?" column, review your findings from the Step 3: Determine the Collections and Conditions of Rules to Create step. If you selected to create custom rules to allow system files to run, note the high-level rule configuration in the column.

Existing Domain, Site, or OU Name	GPO Name	Default rules defined?
OU: Teller-East **OU: Teller-West**	Tellers	Create a path exception to the default rule to exclude \Windows\Temp
OU: HR-All	HR	Use default rule for the Windows path

Step 5: Create a Process for Managing AppLocker Policies and Rules

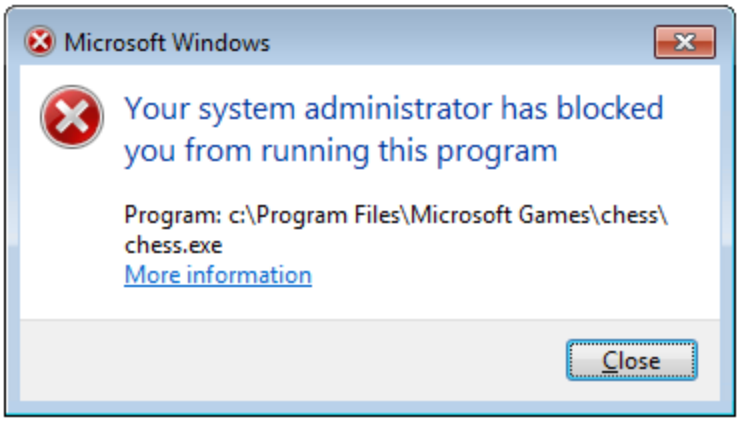
You must consider the following before testing your policy:

- How new rules will be added to the policy
- How existing rules will be updated
- The type of support process will be used for blocked applications
- Whether events will be forwarded for review

User support policy

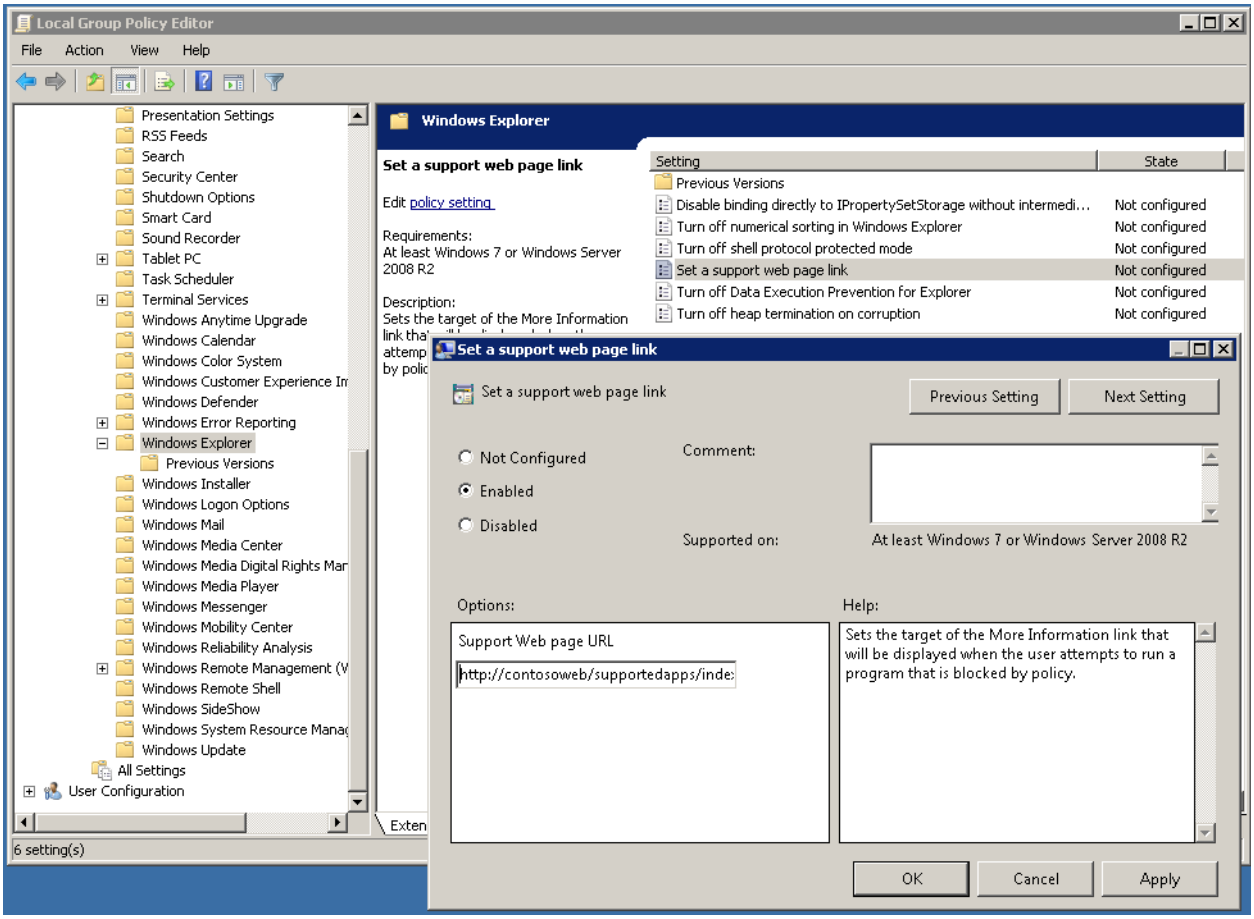
Because AppLocker is preventing unauthorized applications from running, it is important that your organization carefully plan how to provide end user support. First, determine whether you will create an intranet site as a first-level line of support for users who have tried to run a blocked application. AppLocker can be configured to display a message with a custom URL. You can use this URL to redirect users to a support site that you populate with information about why the user received the error (application is blocked) as well as which applications are allowed. If you do not configure AppLocker to display a custom message when an application is blocked, the default message will be displayed.

The following image shows a custom message displayed for the blocked Chess.exe (Microsoft Chess game).



To display a custom URL message when users try to run a blocked application

1. Open the **Group Policy Management** console and navigate to the GPO that you want to edit.
2. Right-click the GPO, and then click **Edit**.
3. In the **Group Policy Management Editor** console, expand **Policies**, expand **Administrative Templates**, expand **Windows Components**, and then select **Windows Explorer**.
4. In the details pane, double-click **Set a support web page link**.
5. Select **Enabled**, and then type the URL of the custom Web page. Click **OK** to apply the setting.



Policy exceptions

You should also determine how your organization will manage exceptions to your written security policy. There might be instances where a user who is blocked from running an application should be granted an exception based upon the organization's policy. For example, if a bank teller at Woodgrove National Bank requires access to Internet Explorer in order to run a Web terminal, the administrator of the Tellers AppLocker policy would have to determine whether to create a custom rule for that user in the GPO or whether to create an Applocker rule on the user's computer.

Event processing

Each time that a process requests permission to run, AppLocker creates an event in the AppLocker event log and lists:

- Which file tried to run
- The file attributes
- The user who initiated the request
- The GUID of the rule that was used to make the AppLocker execution decision.

The AppLocker event log is located in the following path: Applications and Services Logs\Microsoft\Windows\AppLocker. The AppLocker log includes two child logs:

- **EXE and DLL**: Contains events for all files affected by the Executable Rules and DLL Rules collections (.exe, .com, .dll, and .ocx)
- **MSI and Script**: Contains events for all files affected by the Windows Installer Rules and Script Rules collections (.msi, .msp, .ps1, .bat, .cmd, .vbs, and .js)

Collecting these events in a central location can be very helpful for maintaining your AppLocker policy and for troubleshooting rule configuration problems. Event collection technologies, such as those available in Windows 7 and Windows Server 2008 R2, allow administrators to subscribe to specific event channels and have the events from source computers aggregates into a forwarded event log on a Windows Server 2008 R2 collector. For more information about setting up an event subscription, see Configure Computers to Collect and Forward Events (http://technet.microsoft.com/en-us/library/cc748890.aspx).

AppLocker policy maintenance

Because your AppLocker policy will likely manage applications that are updated by the software publisher, you will eventually need to make revisions to your rule set to ensure that the policy is current.

You can edit an AppLocker policy by adding, changing, or removing rules. However, you cannot version the policy by importing additional rules. The safest way to modify an AppLocker policy that is in product is to use Group Policy management software that allows you to version GPOs. An example of this type of software is the Advanced Group Policy Management feature from the Microsoft Desktop Optimization Pack (MDOP). For more information about Advanced Group Policy Management, see the Advanced Group Policy Management Overview (http://www.microsoft.com/downloads/details.aspx?FamilyID=993a34d0-c274-4b46-b9fc-568426b81c5e&DisplayLang=en).

> **Caution**: You should never edit an AppLocker rule set while it is being enforced in Group Policy. Because AppLocker controls what files are allowed run, making simple changes on a live policy can create unexpected behavior.

When a new version of a supported application is deployed

First, you need to determine whether you will continue to support the previous version of the application. If the decision is to withdraw support of the previous version of the application, remove the rule for that application from your rule collection. To add the new version, you must create a new rule for each file that is associated with the application. If you have been using publisher conditions and the version is not specific (e.g., iexplorer.exe rather than iexplorer.exe, version 6.0 and above), then the existing rule(s) might be sufficient to allow the updated file to run. You must ensure, however, that the updated application has not altered the file names or added files to support new functionality. If so, then you must modify the existing rules or make new ones. To continue to reuse a publisher-based rule without a specific file version, you must also ensure that the file's signature is still identical to the previous version—the publisher, product name, and file name (if configured in your rule) must all match in order for the rule to be correctly applied.

To determine whether a file has been modified during an application update, review the publisher's release/update details provided with the update package. You can also review the publisher's Web page to retrieve this information. Each file can also be inspected to determine the version.

For files that are allowed or denied with file hash conditions, you must retrieve the new file hash. To add support for a new version and maintain support for the older version, you can either create a new file hash rule for the new version or edit the existing rule and add the new file hash to the list of conditions.

When a new application is deployed

You must add one or more rules to the existing AppLocker policy in order to support a new application.

When an application is no longer supported

If your organization has determined that it will no longer support an application that has AppLocker rules associated with it, the easiest way to prevent users from running the application from running is to delete the rules.

When an application is blocked but should be allowed

The file could be blocked for three reasons:

- No rule exists to allow the file to run (common)
- An existing rule for the file is too restrictive
- A deny rule is explicitly blocking the file (cannot be overridden)

Before editing the rule set, you must determine what rule is preventing the file from running. You can troubleshoot the problem by using the Test-AppLockerPolicy cmdlet. The topic Test and Update the Policy details how to troubleshoot this type of problem.

Record findings

After determining how your organization will manage your AppLocker policy, you should record your findings.

- **Support policy**: Document the process that you will use for handling calls from users who have attempted to run a blocked application, and ensure that support personnel are well-versed on recommended troubleshooting steps and escalation points for your policy.
- **Event processing**: Document whether events will be collected in a central location, how that store will be archived, and whether the events will be processed for analysis.
- **Policy maintenance**: Detail how rules will be added to the policy, in which GPO that rules should be defined, how to modify rules when applications are retired, versioned, or added.

Step 6: Document Your AppLocker Design

Documenting your AppLocker design decisions will make it easy for the deployment team to implement your design and for the support team to maintain the policy. The appendix of this document includes a master form that you can use to track the different design decisions that you have made while reviewing this document.

Before proceeding, your design document should contain planning information about:

1. What applications in each OU within your organization are in-scope for this project and where to implement AppLocker policies.
2. Your strategy for building rules or using the default rules, rule collections, and rule conditions.
3. How rules will be enforced and how GPOs will be structured for applying AppLocker policy
4. How you intend to provide ongoing support to end users or helpdesk to resolve application access issues
5. The scheme you intend to use to gather and analyze events generated when AppLocker rules are used
6. Your strategy to maintain access to applications and the specific AppLocker rules to enforce that strategy.

Step 7: Create Reference Computers and Automatically Generate Rules

You should create a reference computer for each OU that you will use AppLocker to manage application control policies. While it is not a requirement to use a reference computer to create AppLocker rules, AppLocker includes tools to help you create rules from your existing installed applications.

Perform the following procedures to generate rules for an OU:

1. Create the reference computer
2. Automatically generate rules and create default rules
3. Modify the rule set

Step One: Create the reference computer

A reference computer used for AppLocker policy deployment must be a Windows 7 (Ultimate or Enterprise Editions) or Windows Server 2008 R2 computer that contains the OU structure and corresponding applications to mimic your production environment.

To create a reference computer

1. If the operating system is not already installed, install Windows on the computer.

   Important: To create AppLocker rules on the computer, you must install Windows 7 Ultimate edition, Windows 7 Enterprise edition, or any edition of Windows Server 2008 R2.

2. Install all applications that should be allowed to run in that particular OU.

Step Two: Automatically generate rules and create default rules

On the reference computer, automatically generate rules for the applications that are installed by running the **Automatically Generate Rules** wizard. You will create rules for one rule collection at a time (e.g., you can create executable rules by starting the wizard while in the Executable Rules collection). The wizard analyzes the files that are located in the folder (and its subfolders) that you specify.

You should run the wizard to generate rules for each application that you identified while planning your AppLocker deployment. The Create a List of Applications topic details how to identify the installation folders for different applications.

To create rules for applications installed on a reference computer

1. Log on to the reference computer with an account that is a member of the local Administrators group.
2. Open the Local Security Policy MMC snap-in (secpol.msc).
3. In the console tree, expand **Application Control Policies**, and then expand **AppLocker**.
4. Right-click the rule collection for which you want to create rules and then click **Automatically Generate Rules**.
5. On the **Folder and Permissions** page, specify:
    a. The folder where the application that you want to create rules for is installed.
    b. The user or group that the rule should be applied to. The default is the **Everyone** group.
    c. A name to be prefixed to the names of the rules that will be created for the application. The wizard will automatically create a name based on the folder that you select, but you can also create a custom name.
6. Click **Next**.
7. On the **Rule Preferences** page, specify which order you would like the wizard to create rule types. By default, the wizard will create publisher rules for files that are signed by a publisher and will then create file hash rules for files that are not signed.

    Note: If the Reduce the number of rules created by grouping similar files check box is selected, the wizard will perform the following operations for the different rule types:

    - **Publisher**: One rule is created for all files that have the same publisher and product name.
    - **Path**: One rule is created for the folder that you select. For example, if you select C:\Program Files\ProgramName\ and the files in that folder are not signed, the wizard will create a rule for %programfiles%\ProgramName*
    - **File hash**: One rule is created that contains all of the file hashes. When rule grouping is disabled, the wizard creates a file hash rule for each file.
8. Click **Next**.
9. The **Rule generation progress** dialog box will appear while the files in the directory are being analyzed.
10. The **Review Rules** page lists the results of the file analysis.
    a. Click **Review files that were analyzed** to see which files were reviewed to make rules. On the **Edit Files** page, clearing the check box next to a file's name will prevent a rule from being created for that file.
    b. To see the rules that will be created, click **View rules that will be automatically created**.
11. After you have reviewed the files, click **Create** to create the rules.

Note: If you are running the wizard to create your first rules for a GPO, you will be prompted to create the default rules after completing the wizard. Because the default rules allow critical system files to run, you should always create them during this initial phase. You may edit the default rules at any time. If your organization has decided to edit the default rules or create custom rules to allow the Windows system files to run, ensure that you delete the default rules after replacing them with your custom rules.

Step Three: Modify the rule set

After automatically generating rules, you should review the rules in the MMC details pane. Ensure that the rules are properly scoped to your requirements for the OU for which you are creating AppLocker policy. If a rule is too broad, you should either edit the rule or create an entirely new rule.

To edit a rule

1. In the AppLocker snap-in, click the rule collection that contains the rule you want to edit.
2. In the details pane, right-click the rule you want to edit, and then click Properties.
3. Edit the rule by modifying settings on the tab that aligns to the rule's condition:
    a. **Publisher**: Type new values or enter a * character to specify that any value should be matched.
    b. **Path**: Type a new value for the file or folder path.
    c. **File hash**: To remove a file from the rule, select the file and then click **Remove**. To add a file, click **Browse Files**. To add multiple files, click **Browse Folders**.
4. After completing modifications to the rule, click **OK** or **Apply**.

To create a new rule

1. In the AppLocker snap-in, right-click the rule collection that you want to create the rule for, and then click **Create New Rule**.
2. On the **Before You Begin** page, click **Next**.
3. On the **Permissions** page, select the action (allow or deny) and the user or group that the rule should apply to, and then click **Next**.
4. On the **Conditions** page, select a rule condition, and then click **Next**.
    a. **Publisher**: Click **Browse** to select a signed file, and then user the slider to specify the scope of the rule. To use custom values in any of the fields or to specify a specific file version, select the **Use custom values** check box.
    Note: The slider is disabled when the Use custom values check box is selected.
    b. **Path**: To browse to a file location, select **Browse Files**. To browse to a folder location, select **Browse Folders**.
    Note: When you browse to a file or folder location, the wizard automatically converts absolute file paths to use AppLocker path variables. You may edit the path after browsing to specify an absolute path, or you may type the path directly into the Path text box.
    c. File hash: To add a file, click **Browse Files**.

5. Click **Next**.
6. (Optional) On the **Exceptions** page, specify conditions by which to exclude files from being affected by the rule. Click **Next**.
7. On the **Name and Description** page, either accept the automatically generated rule, or type a new rule name, and then click **Create**.

Step 8: Test and Update the Policy

After creating a rule set, you should test the rules' effectiveness. You should perform these steps for each GPO where you have created AppLocker rules. Because AppLocker rules are inherited from linked GPOs, you should deploy all of the rules for simultaneous testing in all of your test GPOs.

AppLocker testing includes the following steps:

1. Enable the audit only enforcement setting
2. Configure the Application Identity service to start automatically
3. Perform testing on the policy
4. Analyze AppLocker events
5. Modify the policy
6. Repeat steps 3-5 until the policy is complete

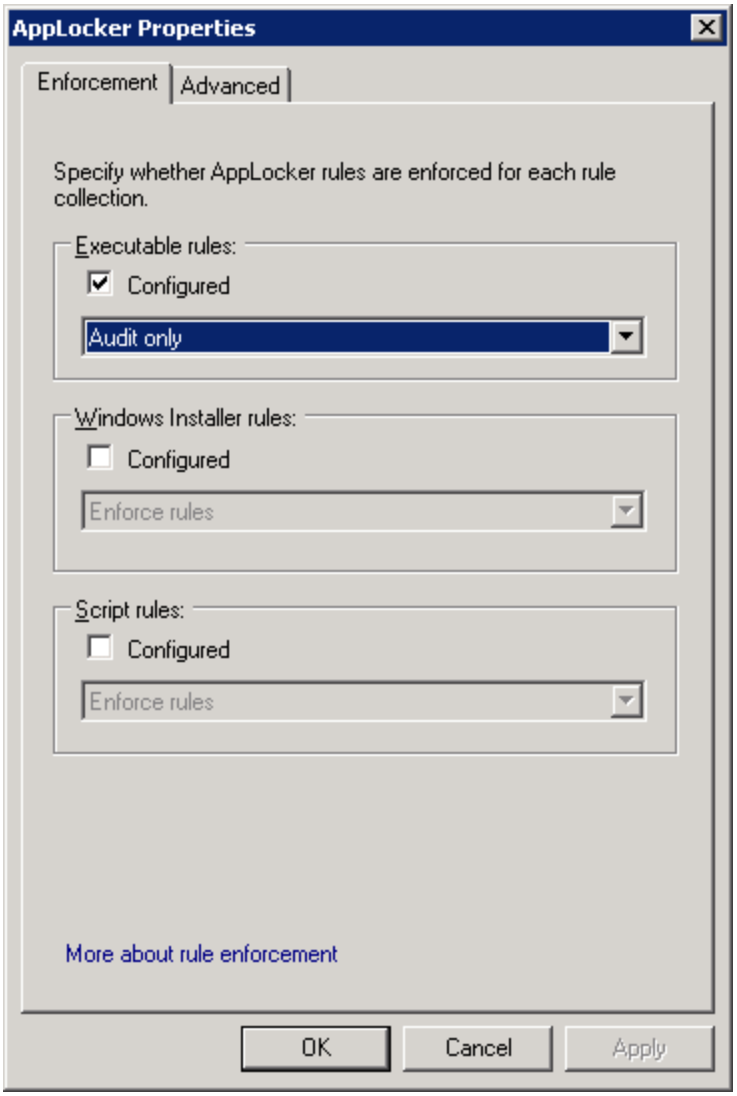
Step One: Enable the audit only enforcement setting

By using the audit only enforcement setting, you can ensure that the AppLocker rules that you have authored are properly scoped for your organization.

To enable the audit only enforcement setting for a GPO

1. Using the **Group Policy Management** console, open the GPO to be edited.
2. In the console tree of the **Group Policy Management Editor**, expand **Computer Configuration**, expand **Policies**, expand **Security Settings**, expand **Application Control Policies**, and then expand **AppLocker**.
3. Right-click **AppLocker**, and then click **Properties**.
4. On the **Enforcement** tab of the **AppLocker Properties** dialog box, select **Configured** for the rule collection that you are editing, and then select **Audit only** from the drop-down list.
5. Click **OK**.

The following screen shot shows the audit only enforcement setting configured for the Executable Rules collection.

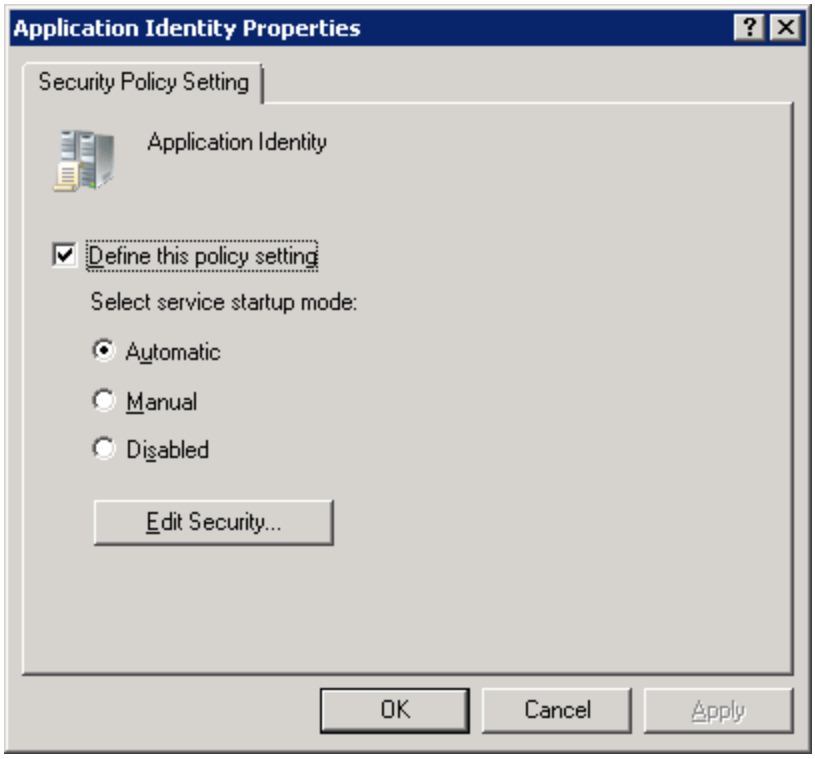


Step Two: Configure the Application Identity service to start automatically

Because AppLocker users the Application Identity service to verify the attributes of a file, you must configure it to start automatically in each GPO that contains AppLocker rules.

To configure the Application Identity service to start automatically

1. Using the **Group Policy Management** console, open the GPO to be edited.
2. In the console tree of the **Group Policy Management Editor**, expand **Computer Configuration**, expand **Policies**, expand **Windows Settings**, expand **Security Settings**, and then click **System Services**.
3. In the details pane, double-click **Application Identity**.
4. In **Application Identity Properties**, configure the service to start automatically.



Step Three: Perform testing on the policy

Test the AppLocker policy to determine if your rule set needs to be modified. Because you have created AppLocker rules, enabled the Application Identity service, and enabled the audit only enforcement setting, the AppLocker should be present on all client computers that receive your AppLocker policy.

The AppLocker PowerShell cmdlet Test-AppLockerPolicy can be used to determine whether any of the rules in your comprehensive rule list will be blocked on your reference computers. Perform the following steps on each reference computer that you used to define AppLocker policy. Ensure that the reference computer is joined to the domain and is receiving AppLocker policy from the appropriate GPO.

To test AppLocker policy with Test-AppLockerPolicy

1. Export the effective AppLocker policy. To do this, you must use the Get-AppLockerPolicy PowerShell cmdlet.
    a. Open a Windows PowerShell V2 prompt window as an administrator.
    b. Before you can use the AppLocker cmdlets, you must import them into PowerShell. To do this, run the following command: `import-module AppLocker`
    c. Use Get-AppLockerPolicy to export the effective AppLocker policy to an XML file:

```
Get-AppLockerPolicy –Effective –XML >
<PathofFiletoExport.XML>
```

2. Use Get-ChildItem to specify the directory that you would like to test, then specify Test-AppLockerPolicy with the XML file from the previous step to test the policy, and use Export-CSV to export the results to a file to be analyzed:

```
Get-ChildItem <DirectoryPathtoReview> -Filter
<FileExtensionFilter> -Recurse | Convert-Path | Test-
AppLockerPolicy –XMLPolicy <PathToExportedPolicyFile> -User
<domain\username> -Filter <TypeofRuletoFilterFor> | Export-CSV
<PathToExportResultsTo.CSV>
```

The following shows example input for Test-AppLockerPolicy:

```
PS C:\ Get-AppLockerPolicy –Effective –XML > C:\Effective.xml

PS C:\ Get-ChildItem 'C:\Program Files\Microsoft Office\' –filter
*.exe –Recurse | Convert-Path | Test-AppLockerPolicy –XMLPolicy
C:\Effective.xml –User contoso\jphillips –Filter
Denied,DeniedByDefault | Export-CSV C:\BlockedFiles.csv
```

In the example, the effective AppLocker policy is exported to the file C:\Effective.xml. The next command then uses Get-ChildItem to recursively gather properties on the .exe files under C:\Program Files\Microsoft Office\. The –XMLPolicy option specifies that the C:\Effective.xml file is an XML AppLocker policy file. By specifying the –User option, you can test the rules for specific users or groups, and the Export-CSV cmdlet allows the results to be exported to a comma-separated file, which can be opened by text and spreadsheet programs like Microsoft Excel.

Step Four: Analyze AppLocker events

You can either manually analyze AppLocker events or use the Get-AppLockerFileInformation PowerShell cmdlet to automate the analysis.

Manually analyzing events

Manually analyzing the events involves viewing the events either in Event Viewer or a text editor and then sorting those events to find patterns. If you have not configured an event subscription, then you will have to review the logs on a sampling of computers in your organization.

Analyzing events with PowerShell

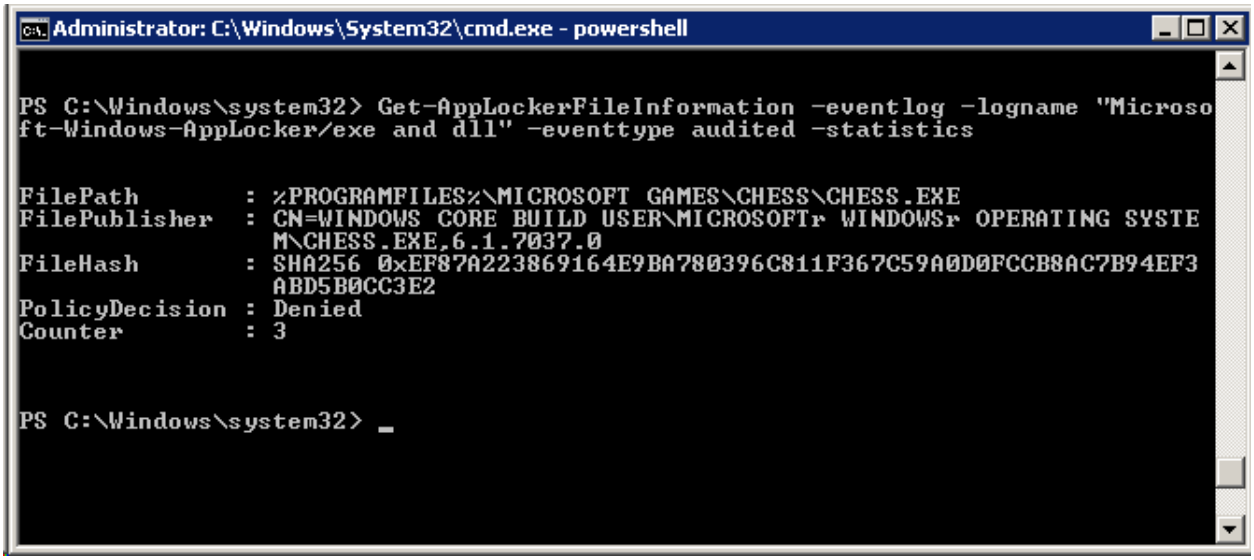
Because PowerShell includes built-in capabilities to remotely administer computers, we recommend that you use the Get-AppLockerFileInformation cmdlet to analyze AppLocker events. If an application is being blocked and should be allowed, you can use the AppLocker PowerShell cmdlets to help troubleshoot the problem.

For both event subscriptions and local events, you can use the **Get-AppLockerFileInformation** PowerShell cmdlet to determine which files have been blocked or would have been blocked (if in the audit only enforcement mode) and how many times the event has occurred for each file. For example,

to see information about how many times that files would have been blocked if enforcement was enabled, you can run:

```
Get-AppLockerFileInformation –EventLog –Logname “Microsoft-
Windows-AppLocker\EXE and DLL” –EventType Audited –Statistics
```

The following image shows the results of running this command for a computer where Chess.exe (Microsoft Chess) has been blocked.



The Counter value shows that the file Chess.exe would have been prevented from running three times if enforcement had been implemented.

For more information about the AppLocker cmdlets and how they can be used to create AppLocker policy from the event log, see the AppLocker PowerShell Technical Reference (LINK).

To review AppLocker events with Get-AppLockerFileInformation

1. Open a Command Prompt window as an administrator.
2. At the command prompt, type `PowerShell`, and then press ENTER.
3. Review how many times a file would have been blocked from running if rules were enforced:

   ```
   Get-AppLockerFileInformation –EventLog –Logname “Microsoft-Windows-
   AppLocker\EXE and DLL” –EventType Audited –Statistics
   ```

   Note: For an event subscription, specify the path to the forwarded event log for the -Logname parameter.

4. Review how many times a file has been allowed to run:

   ```
   Get-AppLockerFileInformation –EventLog –Logname “Microsoft-Windows-
   AppLocker\EXE and DLL” –EventType Allowed –Statistics
   ```

By using Get-AppLockerFileInformation to determine how many times that a file would have been blocked from running, you should review your rule list to determine whether a new rule should be created for the blocked file or whether an existing rule is too strictly defined. Ensure that you check which GPO is currently preventing the file from running. To determine this, you can use the Group Policy results wizard to view rule names.

Step Five: Modify the policy

After you have identified which rules need to be edited or added to the policy, you can use the Group Policy Management Editor to modify the AppLocker rules in the relevant GPOs. For steps that detail how to edit or add a rule, see the Modify the rule set topic in this guide.

Step Six: Repeat steps three through five until the policy is complete

We recommend that you test and update your AppLocker policy several times before enforcing AppLocker rules. After confirming that users would not be prevented from running necessary applications, you can deploy the production enforcement setting.

Step 9: Deploy the Production Enforcement Setting

After you have successfully completed several rounds of testing and modification of the AppLocker policy for each GPO, you are ready to deploy the production enforcement setting. For most organizations, this means switching the AppLocker enforcement setting from audit only to enforce rules. However, it is important to follow the original deployment plan that you created earlier. Depending upon the needs of different business groups in your organization, you might be deploying different enforcement settings for linked GPOs. Refer to your deployment plan for your organization's required production enforcement settings.

Step 10: Maintain the Policy

Now that your AppLocker policy is actively deployed, you can use your rule maintenance plan that you created from the Create a Process for Managing AppLocker Policies and Rules topic.

Editing an AppLocker policy in a production environment

You can edit an AppLocker policy by adding, changing, or removing rules. However, you cannot version the policy by importing additional rules. The safest way to modify an AppLocker policy that is in production is to use Group Policy management software that allows you to version GPOs. An example of this type of software is the Advanced Group Policy Management feature from the Microsoft Desktop Optimization Pack (MDOP). For more information about Advanced Group Policy Management, see the Advanced Group Policy Management Overview (http://www.microsoft.com/downloads/details.aspx?FamilyID=993a34d0-c274-4b46-b9fc-568426b81c5e&DisplayLang=en).

Caution: You should never edit an AppLocker rule set while it is being enforced in Group Policy. Because AppLocker controls what files are allowed run, making simple changes on a live policy can create unexpected behavior.

To edit an AppLocker policy

Note: If you are performing this procedure using Advanced Group Policy Management, check out the GPO before exporting the policy.

1. **Export the policy from the GPO:**
    a. Using the **Group Policy Management** console, open the GPO that you want to edit.
    b. In the console tree of the **Group Policy Management Editor**, expand **Computer Configuration**, expand **Policies**, expand **Windows Settings**, expand **Security Settings**, expand **Application Control Policies**, and then click **AppLocker**.
    c. Right-click **AppLocker**, and then click **Export Policy**.
    d. In the **Export Policy** dialog box, type a name for the exported policy (e.g., the name of the GPO), select a location where to save the policy, and then click **Save**.
    e. The **AppLocker** dialog box will notify you of how many rules were exported. Click **OK**.
2. **Import the policy to a reference computer:**
    a. On the computer where you would like to edit the policy (e.g., a reference computer for the OU), open the **Local Security Policy** editor (secpol.msc).
    b. In the console tree, expand Application Control Policies, and then click AppLocker.
    c. Right-click AppLocker, and then click Import Policy.
    d. In the Import Policy dialog box, locate the file that you exported in step 1d above, and then click **Open**.
    e. The **Import Policy** dialog box will warn you that importing a policy will overwrite the existing rules and enforcement settings. Click **OK** to import and overwrite the policy.
    f. The **AppLocker** dialog box will notify you of how many rules were overwritten and imported. Click **OK**.
3. **Edit the policy:**
    a. To **add** a new rule, right-click the rule collection that you want to edit (i.e., Executable Rules, Windows Installer Rules, Script Rules, or DLL Rules) and then either click **Create New Rule** (to create one rule) or click **Automatically Generate Rules** (to create many rules for a folder).
    b. To **edit** an existing rule, right-click the rule in the details pane and then click **Properties**. Make changes to the rule, and then click **OK**.
    c. To **delete** a rule, right-click the rule in the details pane, click **Delete**, and then confirm that you want to permanently delete the rule.

If you have created multiple AppLocker policies and need to merge them to create one AppLocker policy, you will need to manually merge the policies by hand. You cannot automatically merge one or more policies using the AppLocker snap-in. You must manually edit the policy files to create one rule set.

The AppLocker policy is saved in XML format, and the exported policy can be edited with any text or XML editor. Rule collections are specified within the `RuleCollection Type` element. The XML schema includes four attributes for the different rule collections.

Rule Collection	RuleCollection Type
Executable Rules	Exe
Windows Installer Rules	Msi
Script Rules	Script
DLL Rules	Dll

Rule enforcement is specified with the `EnforcementMode` element. The three enforcement modes in the XML correspond to the three enforcement modes in the AppLocker Group Policy snap-in.

XML Enforcement Mode	Enforcement Mode in Group Policy
NotConfigured	Not configured (rules are enforced)
Disabled	Audit only
Enabled	Enforce rules

Each of the three condition types (i.e., publisher, path, and file hash) use specific elements.

- **Publisher**: The following image is a screen shot of the publisher rule XML format.

```
- <FilePublisherRule Id="f1e6dd1c-83d0-4a21-b9ee-193cc8193090" Name="Windows Live: O=MICROSOFT CORPORATION, L=REDMOND,
  S=WASHINGTON, C=US\WINDOWS LIVE COMMUNICATIONS PLATFORM" Description="" UserOrGroupSid="S-1-1-0" Action="Allow">
  - <Conditions>
    - <FilePublisherCondition PublisherName="O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US"
        ProductName="WINDOWS LIVE COMMUNICATIONS PLATFORM" BinaryName="WLCOMM.EXE">
        <BinaryVersionRange LowSection="14.0.8050.1202" HighSection="14.0.8050.1202" />
      </FilePublisherCondition>
    </Conditions>
  </FilePublisherRule>
```

- **Path**: The following image is a screen shot of the path rule XML format.

```
- <FilePathRule Id="921cc481-6e17-4653-8f75-050b80acca20" Name="(Default Rule) Microsoft Windows
  Program Files Rule" Description="This rule allows 'Everyone' to execute programs in the 'Program Files'
  folder" UserOrGroupSid="S-1-1-0" Action="Allow">
  - <Conditions>
      <FilePathCondition Path="%PROGRAMFILES%\*" />
    </Conditions>
  </FilePathRule>
```

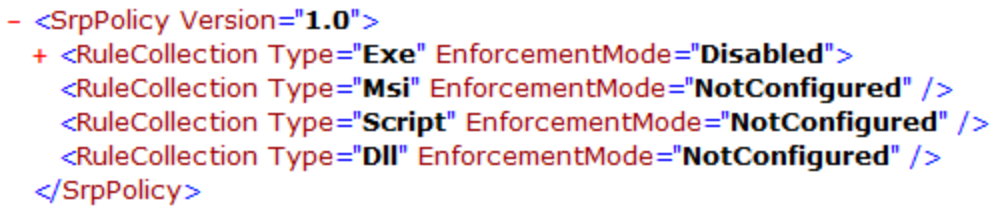
- **File hash**: The following image is a screen shot of the file hash rule XML format.

```
- <FileHashRule Id="75973cd7-4b66-4496-b585-ad1522d6d264" Name="Windows Media Components: WMProEdt.exe" Description=""
  UserOrGroupSid="S-1-1-0" Action="Allow">
  - <Conditions>
    - <FileHashCondition>
        <FileHash Type="Sha256" Data="0x62C138DB048E56ED63A332E87980D9E384BFF2BE8CC2657BD731340A8CF30DFC"
          SourceFileName="WMProEdt.exe" SourceFileLength="32256" />
      </FileHashCondition>
    </Conditions>
  </FileHashRule>
```

To merge two or more AppLocker policies

1. Open the first XML policy file in a text editor or XML editor, such as Windows Notepad.

2. Select the rule collection where you want to copy rules from. The following image shows the four rule collections collapsed in Internet Explorer.



3. Select the rules that you want to add to the other policy file, and then copy the text.
4. Open the second XML policy file where you want to add the copied rules.
5. Select and expand the rule collection where you want to add the rules.
6. At the bottom of the rule list for the collection, after the closing element, paste the rules that you copied from the first policy file. Verify that the opening and closing elements are intact, and then save the policy.
7. Upload the policy to a reference computer and test it to ensure that it is functioning properly.

Appendix: Planning Form

Use the following form to track the planning decisions that you made while designing your AppLocker deployment. Fill out one form for each business group in your organization.

Business group	GroupName
Associated OUs	List of OUs align to each business group
Implement AppLocker?	Yes/No (If no, do not complete the rest of this form for this business group.)
Production enforcement setting	Not configured (default-rules are enforced) / Enforce rules / Audit only
New GPO or Existing?	If using a new GPO for AppLocker, specify the new name. If you are adding the AppLocker rules to an existing GPO, specify the existing name.
Use event forwarding?	Yes/No (If yes, specify tool to be used. If using an event subscription, specify the name of the event collector. This name will be used for getting data with the AppLocker PowerShell cmdlets.)
Custom message?	Yes/No (If yes, specify the URL of the custom Web page)
How will users request access for new application?	Specify process
How will support be given?	Specify process
How will rules be maintained?	Specify process