4. Describe the commond differences between AppLocker and Software Restriction features.

[What is AppLocker?](javascript:void(0))

AppLocker is a feature in Windows Server 2012, Windows Server 2008 R2, Windows 8, and Windows 7 that advances the functionality of the Software Restriction Policies feature. AppLocker contains new capabilities and extensions that reduce administrative overhead and help administrators control how users can access and use files, such as executable files, scripts, Windows Installer files, and DLLs. By using AppLocker, you can:

* Define rules based on file attributes that persist across application updates, such as the publisher name (derived from the digital signature), product name, file name, and file version. You can also create rules based on the file path and hash.
* Assign a rule to a security group or an individual user.
* Create exceptions to rules. For example, you can create a rule that allows all users to run all Windows binaries except the Registry Editor (Regedit.exe).
* Use audit-only mode to deploy the policy and understand its impact before enforcing it.
* Create rules on a staging server, test them, export them to your production environment, and then import them into a Group Policy Object.
* Simplify creating and managing AppLocker rules by using Windows PowerShell cmdlets for AppLocker.

[How does AppLocker differ from Software Restriction Policies?](javascript:void(0))

Software Restriction Policies was originally designed for Windows XP and Windows Server 2003 to help IT professionals limit the number of applications that would require administrator access. With the introduction of User Account Control (UAC) and the emphasis of standard user accounts in Windows Vista®, fewer applications require administrator privileges. As a result, AppLocker was introduced to expand the original goals of Software Restriction Policies (SRP) by allowing IT administrators to create a comprehensive list of applications that should be allowed to run.

For information that compares SRP and AppLocker and explains how you can use them together in the same domain, see [Use AppLocker and Software Restriction Policies in the Same Domain](http://technet.microsoft.com/library/hh994614). For more information about SRP, see the [Software Restrictions Policies Overview](http://technet.microsoft.com/library/hh831534).

[What has changed for AppLocker in Windows Server 2012?](javascript:void(0))

AppLocker can now manage policies for Packaged apps and Packaged app installers. For a complete list of differences in functionality between operating system versions, see the [AppLocker Technical Overview](http://technet.microsoft.com/library/hh831440.aspx)in the Windows Server 2012 TechNet Library. For more information about AppLocker rules for Packaged apps, see [Packaged apps and Packaged app installer rules in AppLocker](http://technet.microsoft.com/library/hh831350). “What types of files can I manage with AppLocker?”

In Windows Server 2008 R2 and Windows 7, you can manage four types of files: executable (.exe), Windows Installer (.msi and .msp), script (.bat, .cmd, .js, .ps1, and .vbs), and DLL (.dll and .ocx). Each of these file types is managed in its own rule collection.

In Windows Server 2012 and Windows 8, in addition to the file types, you can manage .mst and .appx files with AppLocker.

To understand the security issues that are related to how AppLocker manages various files and processes, see [Security Considerations for AppLocker](http://technet.microsoft.com/en-us/library/ee844118(WS.10).aspx).

[How does AppLocker manage Packaged apps?](javascript:void(0))

Typically, an app consists of multiple components: the installer that is used to install the app and one or more .exe file, .dll file, or script. With classic apps, not all these components always share common attributes such as the publisher name, product name, and product version. Therefore, AppLocker controls each of these components separately through the following rule collections: Exe, Dll, Script, and Windows Installers. In contrast, all the components of a packaged app share the same Publisher name, Package name, and Package version attributes. It is therefore possible to control an entire app with a single rule.

For more information about AppLocker rules for Packaged apps, see [Packaged apps and Packaged app installer rules in AppLocker](http://technet.microsoft.com/library/hh831350). For information about how to manage Packaged apps with AppLocker, see[Create a rule for Packaged apps](http://technet.microsoft.com/library/hh994588) and [Add rules for Packaged apps to existing AppLocker rule-set](http://technet.microsoft.com/library/hh994582).

[What are rule conditions?](javascript:void(0))

Rule conditions are properties of files that AppLocker uses to enforce rules. Each AppLocker rule can use one primary rule condition. The following three rule conditions are available in AppLocker:

* **Publisher** rule conditions can only be used for files that are digitally signed by a software publisher. This condition type uses the digital certificate (publisher name and product name) and properties of the file (file name and file version). This type of rule can be created for an entire product suite, which allows the rule in most cases to still be applicable when the application is updated.
* **Path** rule conditions are based on the file or folder installation path of specific applications.
* **File hash** rule conditions are based on the unique file hash that Windows cryptographically computes for each file. This condition type is unique, so each time that a publisher updates a file, you must create a new rule.

[Do AppLocker policies apply to executable files on a portable device?](javascript:void(0))

Yes, AppLocker prevents computers from running unauthorized executable files. It does not protect applications from malicious modifications. This means that the AppLocker rules are applied regardless of where the executable file is located, such as on a network, on a USB drive, or in a mail attachment. For example, if an AppLocker rule exists on the operating system for an application that is hosted on a USB drive and the application is changed by a malicious user, the application is prevented from running on the operating system.

[Does AppLocker check for certificate revocation when defining whether the binary signature is still valid?](javascript:void(0))

Yes, AppLocker does check the certificate revocation while verifying the binary signature. However, if a certificate was allowed and then revoked, there is a 24-hour period before the revocation takes effect.

[Does a deny action on a rule always take precedence?](javascript:void(0))

Yes. However, AppLocker contains a feature that allows you to state exceptions to a deny action on a rule. Rule exceptions, which can be applied to the deny or allow action, permit you to specify files or folders to exclude from the rule. For example, a rule can be created to allow the Everyone group to run any application in the Windows folder except regedit.exe. Another rule can be created to allow the Helpdesk group to run regedit.exe. However, if there was an explicit deny action on regedit.exe, then no other rule permitting the Helpdesk group access to regedit.exe would supersede that rule.

[What security issues should I consider?](javascript:void(0))

Because AppLocker is just part of your comprehensive security strategy, there are certain security issues of which you need to be aware both for planning and operations. For a description of these issues, see[Security Considerations for AppLocker](http://technet.microsoft.com/en-us/library/ee844118(WS.10).aspx).

[Planning and deploying AppLocker policies](javascript:void(0))

[Which editions of the Windows operating systems support AppLocker rules?](javascript:void(0))

**In Windows Server 2008 R2 and Windows 7**

AppLocker rules can be enforced on computers running Windows 7 Ultimate, Windows 7 Enterprise, or any edition of Windows Server 2008 R2 except Windows Web Server 2008 R2 and Windows Server 2008 R2 Foundation.

To create rules for a local computer, the computer must be running Windows 7 Ultimate or Windows 7 Enterprise. If you want to create rules for a Group Policy Object (GPO), you can use a computer that is running any edition of Windows 7 if the Remote Server Administration Tools are installed. AppLocker rules can be created on any edition of Windows Server 2008 R2. Although you can create AppLocker rules on computers running Windows 7 Professional, they will not be enforced on those computers. However, you can create the rules on a computer running Windows 7 Professional and then export the policy for implementation on a computer running an edition of Windows that does support AppLocker rule enforcement.

**In Windows Server 2012 and Windows 8**

AppLocker is supported on all Windows beta evaluation versions except the Server Core installation option.

[What are the policy considerations for Packaged apps?](javascript:void(0))

**Why can I not run my Packaged apps after joining my computer running Windows Server 2012 or Windows 8 to my existing domain?**

By default, if there are no rules in a particular rule collection, AppLocker allows every file that is covered by that rule collection. For example, if there are no Windows Installer rules, AppLocker allows all .msi/.msp/.mst files to run. An existing AppLocker policy that was targeted at computers running Windows Server 2008 R2 and Windows 7 would not have rules for Packaged apps. Therefore, when a computer running Windows Server 2012 or Windows 8 joins a domain where an AppLocker policy is already configured, users would be allowed to run any Packaged app. This might be contrary to your design.

To prevent all Packaged apps from running on a newly domain-joined computer, by default, AppLocker blocks all Packaged apps on a computer running Windows Server 2012 or Windows 8 if the existing domain policy has rules configured in the Exe rule collection. You must take explicit action to allow Packaged apps in your enterprise. If you want to allow all Packaged apps, then you can create a default rule for the Packaged apps rule collection, or you can choose to allow only a select set of Packaged apps.

**Why can I not create hash- or path-based rules for Packaged apps?**

All Packaged apps and Packaged app installers are signed by the publisher of the package. In contrast, classic apps are not always signed; and therefore, AppLocker supports hash- or path-based rules. For more information about Packaged apps and Packaged app installer rules, see [Packaged apps and Packaged app installer rules in AppLocker](http://technet.microsoft.com/library/hh831350.aspx).

**Why are all the Packaged apps not listed in the application inventory wizard?**

Certain application packages are framework packages that are leveraged by other apps. By themselves, these packages cannot do anything, but blocking such packages can inadvertently cause apps that you want to allow to fail. You can create Allow or Deny rules for the Packaged apps that use these framework packages. The AppLocker user interface deliberately filters out all the packages that have registered themselves as framework packages.

[Can Software Restriction Policies’ rules be migrated to AppLocker rules?](javascript:void(0))

No, not directly. AppLocker rules are not based on the same technology as Software Restriction Policies’ rules. You should carefully analyze your existing Software Restriction Policies’ rules and determine how they would conceptually map to new AppLocker rules.

For information that compares SRP and AppLocker, and explains how you can use them together in the same domain, see [Use AppLocker and Software Restriction Policies in the Same Domain](http://technet.microsoft.com/library/hh994614). For more information about SRP, see the [Software Restrictions Policies Overview](http://technet.microsoft.com/library/hh831534).

[Can I configure the default run level with AppLocker? SRP allowed me to set the default run level to “Always allow” and then create Deny rules.](javascript:void(0))

The default rule action is to run only those applications that are specifically allowed in the AppLocker rule collection. While there is no setting to configure the default rule behavior, you can use AppLocker's default rule collection to override the default rule behavior. To do this, create an allow rule with a path condition set to \*. This configuration will allow all files to run. You can then create deny rules to block specific files.

Specifying files that are allowed is a more secure and manageable way to control applications than specifying files that are denied. Although initial rule setup can be easier with a Deny list, you will need many more rules to block specific files than if you create a list of allowed files.

[Can I test my AppLocker rules before I enforce them?](javascript:void(0))

Yes, you can use the AppLocker **Audit only** enforcement mode to test your rules before they are enforced. When implemented, applications that would have been blocked will be recorded as warning events in the AppLocker logs.

[Can AppLocker rules be applied to specific users or groups?](javascript:void(0))

Yes, rules can be created for specific users or groups. However, a rule can only apply to one user or one group. You can also create AppLocker rules to apply to all users (the Everyone group) and then apply that GPO to a specific computer group.

[Do I need to upgrade my domain controllers to use AppLocker rules?](javascript:void(0))

No, you do not need to update your domain controllers. You can use existing domain controllers running Windows Server 2003 or Windows Server 2008 to host the AppLocker policy. However, you cannot use computers running Windows Server 2003 or Windows Server 2008 to create AppLocker rules.

[Does AppLocker use any services for its rule enforcement?](javascript:void(0))

Yes, AppLocker uses the Application Identity service (AppIDSvc) for rule enforcement. For AppLocker rules to be enforced, this service must be set to start automatically in the GPO.

[How do I create DLL rules?](javascript:void(0))

To create DLL rules in a GPO, you must enable the DLL rule collection. (The DLL rule collection is disabled by default).

[Why is the DLL rule collection disabled by default?](javascript:void(0))

Managing DLLs can be a difficult task. Each application requires that specific DLLs are allowed to run, and one application can start many DLLs. For this reason, implementing DLL rules is a more advanced way of using AppLocker. Improperly configuring DLL rules can cause application compatibility problems, and because AppLocker checks whether a DLL is allowed each time before it is allowed to run, many AppLocker events can be generated in the event log. Therefore, you should carefully plan your DLL rules before enabling the rule collection.

[Can you block all applications except from a certain software publisher?](javascript:void(0))

Yes. You can do this by creating a publisher condition rule that allows all files to run that are signed by the specific software publisher. In some cases for binaries that are created dynamically, you could create a path rule condition.

|  |
| --- |
| **noteNote** |
| Use the **Audit only** enforcement mode to verify that your application list is comprehensive before enforcing the rules. |

[Will performance be affected at application startup?](javascript:void(0))

Users generally will not notice any difference in application startup time when AppLocker rules are enforced. As with most policies, there is an added overhead to either check or set the policy. When enforcing a DLL rule collection, performance at application startup might be degraded if the application loads numerous DLLs because AppLocker checks each DLL.

[Can AppLocker control Application Virtualization applications?](javascript:void(0))

Yes, but there are some considerations:

* If AppLocker rules are enabled, a path rule condition with an allow action must be set for the Application Virtualization (App-V) installation path.
* If App-V is installed in the default location, generating the default AppLocker rules will be sufficient to allow it to run.

[What is the maximum number of rules AppLocker can evaluate?](javascript:void(0))

There is no limit to the number of rules you can create, but consider that performance could degrade with more rules to evaluate and enforce. This is the same whether the rules are applied through a local security policy or through a GPO. The more rules per GPO, the longer AppLocker requires for evaluation. The number of AppLocker rules in a GPO is capped by the maximum supported size of the GPO which is 100 MB. However, the number of rules that can fit in 100 MB can vary by the type of rules.

[Managing AppLocker policies](javascript:void(0))

[Why are my AppLocker rules not being enforced?](javascript:void(0))

The most common reasons why the AppLocker rules might not be enforced are:

* The Application Identity service (AppIDsvc) is not running.
* Rule enforcement is set to **Audit only**.

[How can I view the AppLocker events on client computers?](javascript:void(0))

To view AppLocker events, you can use event forwarding technologies, Event Viewer (eventvwr.msc), or the **Get-WinEvent** Windows PowerShell cmdlet.

You can either use Remote Desktop to log on to a client computer or physically log on to that computer to view or collect the AppLocker events.

In Event Viewer, AppLocker events are stored in a log under: **Applications and Services Logs\Microsoft\Windows\AppLocker**. There are two child logs: one for executable files and DLLs and another for Windows Installer files and scripts.

[What should I do if my rules are too restrictive and Windows is functioning irregularly or will not start?](javascript:void(0))

When AppLocker is enabled, only applications that are specified will be allowed to run. When you first create rules, AppLocker will prompt you to create the default rules. These default rules ensure that key Windows system files and all files in the Program Files directory will be permitted to run. While the default rules are not mandatory, we recommend that you start with the default rules as a baseline and then edit them or create your own to ensure that Windows will function properly.

If computers cannot start properly due to your AppLocker policy, edit the AppLocker rules in the corresponding GPO to be less restrictive. If the AppLocker rules are defined in a computer's local policy, start the computer in Safe Mode, create the default AppLocker rules, and then restart the computer.

[Can I control AppLocker through the command line?](javascript:void(0))

Yes. AppLocker provides Windows PowerShell cmdlets designed to streamline the administration of an AppLocker policy. They can be used to help create, test, maintain, and troubleshoot an AppLocker policy. The cmdlets are intended to be used in conjunction with the AppLocker user interface that is accessed through the Microsoft Management Console (MMC) snap-in extension to the Local Security Policy snap-in and Group Policy Management Console.

[Can AppLocker be managed through a virtual machine?](javascript:void(0))

No. A virtual machine is a separate image. Therefore, the image cannot access the policy files of the computer that hosts the AppLocker policies.

[Can emergency management tasks, such as applying a policy exception to a single user, be done from a remote desktop?](javascript:void(0))

Some tasks can be done by using Remote Desktop. AppLocker policies can be applied by using domain GPOs, local GPOs, or both. If a user requests access to an application, you can use one of the Windows PowerShell cmdlets, or you can use the Local Security Policy snap-in (secpol.msc) to add a local rule to temporarily allow that application. In both cases, you need to have administrator privileges. You can either do this by using Remote Desktop or by using the Windows PowerShell remote access capabilities.

[Can an administrator undo or correct a policy if a mistake is made?](javascript:void(0))

AppLocker does not provide a way to undo an action. However, there is a slight delay in the application of policies, so you can change your rules if you still have the AppLocker snap-in open. The policy will take some time to propagate to a client computer that is joined to a domain, so you might be able delete the erroneous rules and create the correct ones before the policy is applied.

|  |
| --- |
| **CautionCaution** |
| You should not edit an AppLocker rule collection while it is being enforced in Group Policy. Because AppLocker controls what files are allowed to run, making changes to a live policy can create unexpected behavior. |

[Can I back up my permissions or rules?](javascript:void(0))

Yes, there are three ways of doing this:

* Back up the GPOs by using the GPMC (for domain policies only).
* Export the AppLocker policy by using the AppLocker snap-in.
* Create a script by using the **Get-AppLockerPolicy** Windows PowerShell cmdlet to export the policy.

[Does AppLocker allow for time segments during the day when an application can be used?](javascript:void(0))

No, AppLocker cannot be used to allow applications to run at certain times during the day or limit the number of hours per day that an application can run.

[Why can only an administrator change AppLocker policies?](javascript:void(0))

This prevents any standard user that is logged on to a computer from modifying the AppLocker rules to access or add an application. On a computer that is joined to a domain, the computer's administrator can create AppLocker rules that could be merged with a domain-level rules as stated in the domain GPO.

[Can AppLocker rules be created to allow certain computer users to access a program and deny access to others?](javascript:void(0))

Yes, you can target AppLocker rules to users and groups. You can create as many rules as you want for the same application. For example, you could have one rule that allows the Finance group to run winword.exe, and you could also have a second rule that allows the HR group to run winword.exe.

[How can I temporarily allow a user to run or install applications?](javascript:void(0))

There are a variety of methods, and the best one will depend on your administrative practices. The following are some possible methods:

* You can set the enforcement mode on the relevant rule collection to **Audit only** so AppLocker will not block any application for the present time. Then, you can change the enforcement mode to **Enforce rules** when you are ready.
* You can create an organizational unit (OU) that has a separate set of rules but does not block the users from running a particular application. Move the user to this OU temporarily while they install the update or application. Then, move them back to the OU where the original rule enforcement occurs.

[Can AppLocker detect rule conflicts?](javascript:void(0))

No. However, you can use the Windows PowerShell cmdlets **Get-AppLockerPolicy** and **Test-AppLockerPolicy** to check whether specific files are allowed based on an AppLocker policy. A second option is to create a duplicate policy (for example, by using a reference computer), which then allows you to test the policy before deployment and verify that it provides the intended results.

[Do you need a domain and Group Policy infrastructure to use AppLocker policies?](javascript:void(0))

No. You can run AppLocker on a stand-alone computer running Windows 8, Windows 7 Ultimate or Windows 7 Enterprise. You can distribute rules to those computers by using AppLocker's export/import feature.

[What happens if a certificate expires for a file that I have set a publisher rule for? Does AppLocker support time stamping?](javascript:void(0))

If the application's certificate expires while the rule is enforced, the binary file will be blocked from running. A binary file is considered signed as long as the timestamp happened during the validity period of both the signing of the certificate and the time stamping of the certificates in the certificate chain.

<http://technet.microsoft.com/en-us/library/ee619725(v=ws.10).aspx#BKMK_SRPdifferences>