



# Using Software Restriction Policies to Protect Against Unauthorized Software

Published: January 1, 2002 | Updated: May 25, 2004

## Abstract

Software restriction policies are a new feature in Microsoft® Windows® XP and Windows Server 2003. This important feature provides administrators with a policy-driven mechanism for identifying software programs running on computers in a domain, and controls the ability of those programs to execute. Software restriction policies can improve system integrity and manageability—which ultimately lowers the cost of owning a computer.



## On This Page

- ↓ [Introduction](#)
- ↓ [Software Restriction Policies—An Overview](#)
- ↓ [Software Restriction Policy Architecture](#)
- ↓ [Software Restriction Policy Options](#)
- ↓ [Software Restriction Policy Design](#)
- ↓ [Step-by-Step Guide for Designing a Software Restriction Policy](#)
- ↓ [Step-by-Step Guide for Creating Additional Rules](#)
- ↓ [Commonly Overlooked Rules](#)
- ↓ [Scenarios](#)
- ↓ [Deployment Considerations](#)
- ↓ [Troubleshooting Software Restriction Policies](#)
- ↓ [Appendix](#)
- ↓ [Summary](#)
- ↓ [Related Links](#)

## Introduction

Software restriction policies are a part of Microsoft's security and management strategy to assist enterprises in increasing the reliability, integrity, and manageability of their computers. Software restriction policies are one of many new management features in Windows XP and Windows Server 2003.

This article provides an in-depth look at how software restriction policies can be used to:

- Fight viruses
- Regulate which ActiveX controls can be downloaded
- Run only digitally signed scripts
- Enforce that only approved software is installed on system computers
- Lockdown a machine

## Expanded Management Capabilities

Windows 2000 brought significant management capabilities to the Windows platform. In Windows 2000, you could manage the software for your machines in the following ways:

- Application settings allowed you to customize an application once through Group Policy, and then distribute that customization to all domain users who required it.
- The Software Installation snap-in provided a means to centrally manage software distribution in your organization. When the user selected an application from the Start menu for the first time, it set up automatically, and then opened. You could also publish applications to groups of users, making the application available for users to install.
- Security settings defined a security configuration within a Group Policy Object (GPO). Security configuration consisted of settings for: account policies, local policies, event log, registry, file system, public key policies, and other policies.

**Windows XP and Windows Server 2003** expand the management capabilities of Windows 2000 by adding the following features:

- **Better diagnostic and planning information** through Resultant Set of Policies (RSOP). For more information, see the article [Windows 2000 Group Policy](#)
- **Ability to use Windows Management Instrumentation (WMI) filtering.** In Windows 2000 you could apply policies based on organizational information in Active Directory®. In Windows XP you can use WMI information to apply group policies to, for example, machines with a certain build or service pack level of Windows.

Software restriction policies integrate with the operating system and common scripting runtimes to control the running of software at execution. In Windows 2000 you could hide access to applications by removing them from the Start menu or hiding the Run command. New software restriction policies go beyond this by simply removing the common access point for software.

[↑ Top of page](#)

## Software Restriction Policies—An Overview

This section discusses the behavior of hostile code and problems associated with unknown code.

### Hostile Code Has More Ways to Get In

With the increased use of networks and the Internet in daily business computing, the potential for encountering hostile code is higher than ever before. People collaborate in more sophisticated ways by using e-mail, instant messaging, and peer-to-peer applications. As these collaboration opportunities increase, so does the risk of viruses, worms, and other hostile code invading your systems. Remember: e-mail and instant messaging can transport unsolicited hostile code. Hostile code can take many forms. It can range from native Windows executables (.exe), to macros in word processing documents (.doc), to scripts (.vbs).

Viruses and worms often use social engineering to trick users into activating them. With the sheer number and variety of forms that code can take, it can be difficult for users to know what is safe to run and what is not. When activated, hostile code can damage content on a hard disk, flood a network with a denial-of-service attack, send confidential information out to the Internet, or compromise the security of a machine.

### The Problem with Unknown Code

Hostile code is not the only threat—many non-malicious software applications also cause problems. Any software not known and supported by an organization can conflict with other applications or change crucial configuration information. Software restriction policies were designed to help organizations control not just hostile code, but any unknown code—malicious or otherwise.

### Responding to Unknown Code

Software restriction policies help a business respond to unknown code by:

- Providing a way to define a list of what is trusted code versus what is not.
- Providing a flexible, policy-based approach for regulating scripts, executables, and ActiveX controls.

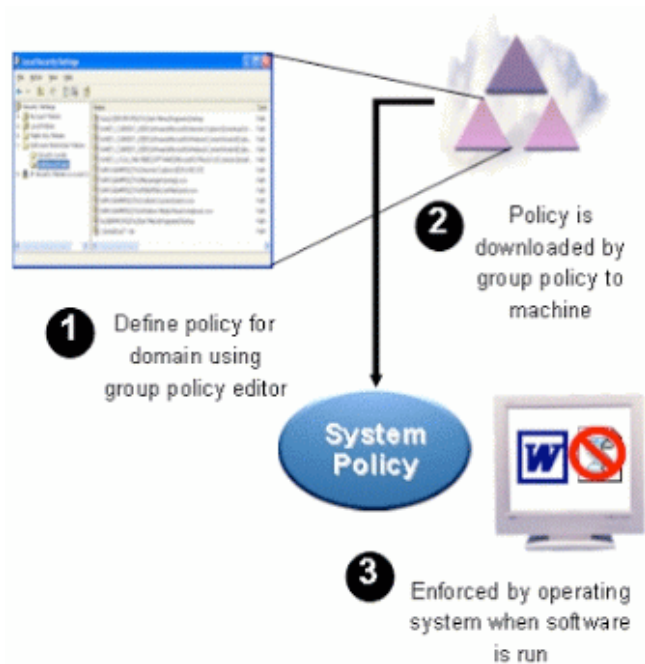
- Enforcing the policy automatically.

[↑ Top of page](#)

## Software Restriction Policy Architecture

Figure 1 below shows the three components of a software restriction policy:

1. An administrator creates the policy by using the Group Policy Microsoft Management Console (MMC) snap-in for a particular Active Directory container site, domain, or organizational unit.
2. The policy is downloaded and applied to a machine. User policies apply the next time a user logs on. Machine policies apply when a machine starts up.
3. When a user starts a program or script, the operating system or scripting host checks the policy and enforces it.



[See full-sized image.](#)

### Unrestricted or Disallowed

A software restriction policy is created using the MMC Group Policy snap-in. A policy consists of a default rule about whether programs are allowed to run, and exceptions to that rule. The default rule can be set to Unrestricted or Disallowed—essentially run or don't run.

Setting the default rule to Unrestricted allows an administrator to define exceptions; for example, the set of programs that are not allowed to run. A more secure approach is to set the default rule to Disallowed and specify only the programs that are known and trusted to run.

### Default Security Level

There are two ways to use software restriction policies:

- **If an administrator knows all of the software that should run**, then a software restriction policy can be applied to control execution to only this list of trusted applications.

- **If all the applications that users might run are not known**, then administrators can step in and disallow undesired applications or file types as needed.

### Four Rules Identify Software

The purpose of a rule is to identify one or more software applications, and specify whether or not they are allowed to run. Creating rules largely consists of identifying software that is an exception to the default rule. Each rule can include descriptive text to help communicate why the rule was created.

A software restriction policy supports the following four ways to identify software:

- **Hash**—A cryptographic fingerprint of the file.
- **Certificate**—A software publisher certificate used to digitally sign a file.
- **Path**—The local or universal naming convention (UNC) path of where the file is stored.
- **Zone**—Internet Zone

### Hash Rules

A hash rule is a cryptographic fingerprint that uniquely identifies a file regardless of where it is accessed or what it is named. An administrator may not want users to run a particular version of a program. This may be the case if the program has security or privacy bugs, or compromises system stability. With a hash rule, software can be renamed or moved into another location on a disk, but it will still match the hash rule because the rule is based on a cryptographic calculation involving file contents.

A hash rule consists of three pieces of data, separated by colons:

- MD5 or SHA-1 hash value
- File length
- Hash algorithm ID

It is formatted as follows:

```
[MD5 or SHA1 hash value]:[file length]:[hash algorithm id]
```

Files that are digitally signed will use the hash value contained in the signature, which may be SHA-1 or MD5. Files that are not digitally signed will use an MD5 hash.

**Example:** The following hash rule matches a file with a length of 126 bytes and with contents that match the MD5 (denoted by the hash algorithm identifier of 32771) hash of 7bc04acc0d6480af862d22d724c3b049—

```
7bc04acc0d6480af862d22d724c3b049:126:32771
```

### Certificate Rules

A certificate rule specifies a code-signing, software publisher certificate. For example, a company can require that all scripts and ActiveX controls be signed with a particular set of publisher certificates. Certificates used in a certificate rule can be issued from a commercial certificate authority (CA) such as VeriSign, a Windows 2000/Windows Server 2003 PKI, or a self-signed certificate.

A certificate rule is a strong way to identify software because it uses signed hashes contained in the signature of the signed file to match files regardless of name or location. If you wish to make exceptions to a certificate rule, you can use a hash rule to identify the exceptions.

### Path Rules

A path rule can specify a folder or fully qualified path to a program. When a path rule specifies a folder, it matches any program contained in that folder and any programs contained in subfolders. Both local and UNC paths are supported.

**Using Environment Variables in Path Rules.** A path rule can use environment variables. Since path rules are evaluated in the client environment, the ability to use environment variables (for example, %WINDIR%) allows a rule to adapt to a particular user's environment.

**Important:** Environment variables are not protected by access control lists (ACL). If users can start a command prompt they can redefine an environment variable to a path of their choosing.

**Using Wildcards in Path Rules.** A path rule can incorporate the '?' and '\*' wildcards, allowing rules such as "\*.vbs" to match all Visual Basic® Script files. Some examples:

- "\\DC-??\login\$" matches \\DC-01\login\$, \\DC-02\login\$
- "\*\Windows" matches C:\Windows, D:\Windows, E:\Windows
- "c:\win\*" matches c:\winnt, c:\windows, c:\windir

**Registry Path Rules.** Many applications store paths to their installation folders or application directories in the Windows registry. You can create a path rule that looks up these registry keys. For example, some applications can be installed anywhere on the file system. These locations may not be easily identifiable by using specific folder paths, such as C:\Program Files\Microsoft Platform SDK, or environment variables, such as %ProgramFiles%\Microsoft Platform SDK. If the program stores its application directories in the registry, you can create a path rule that will use the value stored in the registry, such as %HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\PlatformSDK\Directories\Install Dir%.

This type of path rule is called a registry path rule. The registry path is formatted as follows:

%[Registry Hive]\[Registry Key Name]\[Value Name]%

**Note:** Any registry path rule suffix should not contain a \ character immediately after the last % sign in the rule.

- The registry path must be enclosed in percent signs ("%").
- The registry value must be a REG\_SZ or REG\_EXPAND\_SZ. You cannot use HKLM as an abbreviation for HKEY\_LOCAL\_MACHINE, or HKCU as an abbreviation for HKEY\_CURRENT\_USER.
- If the registry value contains environment variables, these will be expanded when the policy is evaluated.
- A registry path rule can also contain a suffix path such as %HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Cache%OLK\* This registry path rule identifies the folder that Microsoft Outlook XP uses to store attachments before launching them. The attachment folder always starts with the letters "OLK" so the rule uses wildcard matching. As an example, this rule matches the following path: C:\Documents and Settings\username\Local Settings\Temporary Internet Files\OLK4

**Important** When you set a path rule, you should check the access control list (ACL) entries on the path. If users have write access to a path, they can modify its contents. For example, if you allow C:\Program Files, any power user on the machine can copy software into the Program Files folder.

**Path Rule Precedence.** When there are multiple matching path rules, the most specific matching rule takes precedence

The following is a set of paths, from highest precedence (more specific match) to lowest precedence (more general match).

- Drive:\Folder1\Folder2\FileName.Extension
- Drive:\Folder1\Folder2\\*.Extension
- \*.Extension
- Drive:\Folder1\Folder2\

- Drive: \Folder1\

### Zone Rules

A rule can identify software from the Internet Explorer zone from which it is downloaded. These zones are:

- Internet
- Intranet
- Restricted Sites
- Trusted Sites
- My Computer

Currently this applies to only Windows Installer (\*.MSI) packages. It does not apply to software downloaded in Internet Explorer.

### When to Use Each Rule

**Note:** Each rule has a globally unique identifier (GUID) associated with it. An example GUID is {f8c2c158-e1af-4695-bc93-07cbefbdc594}. Two identical rules will have two different GUIDs. GUIDs help you troubleshoot to determine the specific rule in the specific policy that is being used. See the Troubleshooting section later in this article for more information.

**Table 1 When to Use Each Rule**

Task	Recommended Rule
You want to allow or disallow a specific version of a program	<b>Hash rule</b> Browse to file to create hash
You want to identify a program that is always installed in the same place	<b>Path rule with environment variables</b> %ProgramFiles%\Internet Explorer\iexplore.exe
You want to identify a program that can be installed anywhere on client machines	<b>Registry path rule</b> %HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\InoculateIT\6.0\Path\HOME%
You want to identify a set of scripts on a central server	<b>Path rule</b> \\SERVER_NAME\Share
You want to identify a set of scripts on a set of servers, DC01, DC02, and DC03	<b>Path rule with wildcards</b> \\DC??\Share
You want to disallow all .vbs files, except those in a login script directory	<b>Path rule with wildcards</b> *.VBS set to Disallowed \\LOGIN_SRV\Share\*.VBS set to Unrestricted
You want to disallow a file installed by a virus that is always called flcss.exe	<b>Path rule</b> flcss.exe, set to Disallowed

**Table 1 When to Use Each Rule**

Task	Recommended Rule
You want to identify a set of scripts that can be run anywhere	<b>Certificate rule</b> Certificate used to digitally sign the scripts
You want to allow software to be installed from trusted Internet zone sites	<b>Zone rule</b> Trusted Sites set to Unrestricted

**Rule Precedence**

Rules are evaluated in a specific order. The rules that more specifically match a program win over rules that more generally match a program.

- Hash rule
- Certificate rule
- Path rule
- Internet zone rule
- Default rule

Table 2 and the following examples illustrate how rules are processed when a program is started.

**Table 2 Understanding Rule Precedence**

Default Security Level: Unrestricted		
Hash Rules		
Rule 1	Hash of pagefileconfig.vbs	Disallowed
Certificate Rules		
Rule 2	IT Management Certificate	Unrestricted
Path Rules		
Rule 3	%WINDIR%\System32\*.VBS	Unrestricted
Rule 4	*.VBS	Disallowed
Rule 5	%WINDIR%	Unrestricted

Program being started: C:\WINDOWS\SYSTEM32\EventQuery.vbs

This program matches the following rules:

- Rule 3 because it is a .vbs file in the System32 folder.
- Rule 4 because it has a .vbs extension.
- Rule 5 because it is stored in a subfolder of the Windows directory.

Rule 3 is the most specific match for this program. Because Rule 3 has a security level of Unrestricted, the program is allowed to run.

Program being started: C:\WINDOWS\SYSTEM32\pagefileconfig.vbs

This program matches the following rules:

- Rule 1 because the hash in the rule matches the hash of the file.
- Rule 3 because it is a .vbs file in the System32 folder.
- Rule 4 because it has a .vbs extension.
- Rule 5 because it is stored in a subfolder of the Windows directory.

Rule 1 is the most specific match for this program. Because Rule 1 has a security level of Disallowed, the program is disallowed.

Program being started: \\LOGIN\_SRV\Scripts\CustomerScript1.vbs

This program matches the following rules:

- Rule 2 because it is digitally signed by the certificate belonging to the customer's IT management group.
- Rule 4 because it has a .vbs extension.

Rule 2 is the most specific match for this program. Because Rule 2 has a security level of Unrestricted, the program is allowed to run.

Program being started: C:\Documents and Settings\user1\LOVE-LETTER-FOR-YOU.TXT.VBS

This program matches Rule 4 because it has a .vbs extension.

Rule 4 is the most specific match for this program. Because the Rule 4 has a security level of Disallowed, the program is disallowed.

[↑ Top of page](#)

## Software Restriction Policy Options

This section discusses the various options that influence the behavior of a software restriction policy. These options alter the scope of enforcement behavior or the Authenticode trust settings for digitally signed files.

### Enforcement Options

There are two enforcement options: DLL checking and Skip Administrators.

#### DLL Checking

A program, such as Internet Explorer consists of an executable file, iexplore.exe, and many supporting dynamic link libraries (DLL). By default, software restriction policy rules are not enforced against DLLs. This is the recommended option for most customers for three reasons.

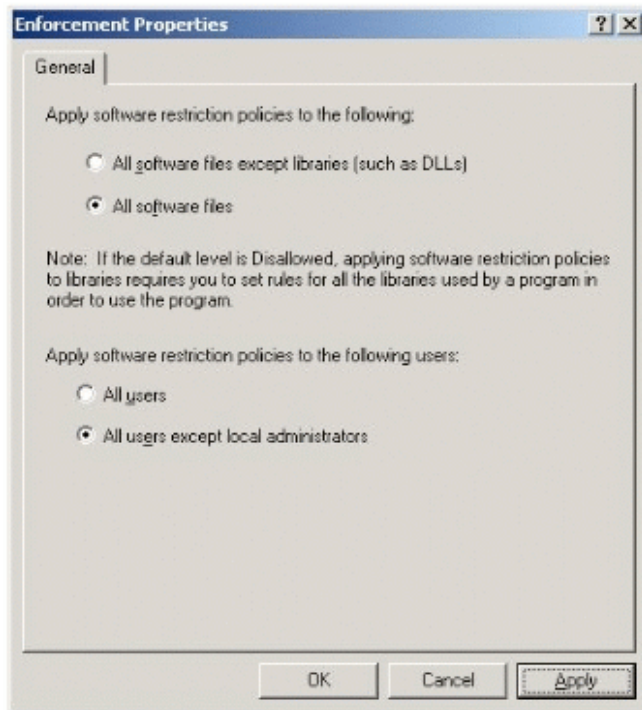
- Disallowing the main executable file prevents the program from running, so there is no need to disallow all of the constituent dynamic link libraries.
- DLL checking results in performance degradation. If a user runs 10 programs during a logon session, the software restriction policy is evaluated 10 times. If DLL checking is turned on, the software restriction policy is evaluated for each DLL load within each program. If each program uses 20 DLLs, this results in 10 executable program checks plus 200 DLL checks, so the software restriction policy is evaluated 210 times.
- If the default security level is set to Disallowed, then not only does the main executable file have to be identified to allow it to run, but all of its constituent DLLs also must be identified, which can be burdensome.

DLL checking is provided as an option for environments that want the highest assurance possible when running programs. While viruses primarily target executables for infection, some target DLLs. To ensure that a program has not been infected by a virus, you can use a set of hash rules that identify the executable and all of its required DLLs.

To turn on DLL checking:

- Select the following option in the **Enforcement Properties** dialog box, as shown in Figure 2 below:

**Apply software restriction policies to the following > All software files**



**Figure 2: Setting Enforcement Properties**

[See full-sized image.](#)

### Skip Administrators

An administrator may want to disallow the running of programs for most users, but allow administrators to run anything. For example, a customer may have a shared machine that multiple users connect to using Terminal Server. The administrator may want users to be able to run only specific applications on the machine, but allow members of the local administrators group to run anything. To do this, use the **Skip Administrators** option.

If the software restriction policy is created in a GPO attached to an object in Active Directory, the preferred way to skip administrators is to deny the **Apply Group Policy** permission on the GPO to a group containing the administrators. This way less network traffic is consumed downloading GPO settings that do not apply to administrators. However, software restriction policies defined in Local Security Policy objects have no way to filter based on users. In this case the Skip Administrators option should be used.

To turn on Skip Administrators:

- Select the following option in the **Enforcement Properties** dialog box as shown in Figure 2 above:

**Apply software restriction policies to the following users > All users except local administrators**

**Note:** Setting the Skip Administrators option is only valid for machine policies.

### Defining Executables

**The Designated File Types** dialog box shown in Figure 3 below lists the file types to which the software restriction policy applies. The designated file types are file types that are considered executable. For example, a screen saver file (SCR), is considered executable because when double-clicked in Windows Explorer it is loaded as a program.

The rules in a software restriction policy only apply to the file types listed in the Designated File Types dialog box. If you

environment uses a file type that you want to be able to set rules on, add it to the list. For example, if you use Perl scripting files, you may choose to add .pl and other file types associated with the Perl engine to the Designated File Types list.

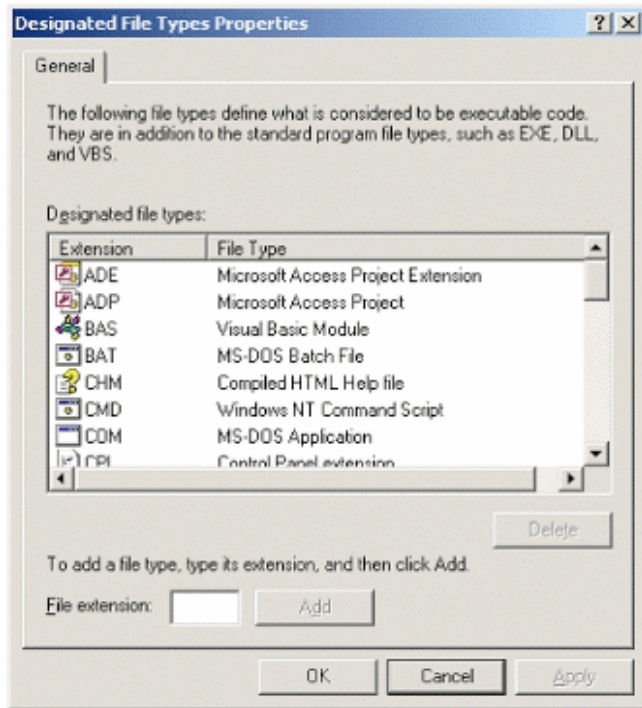
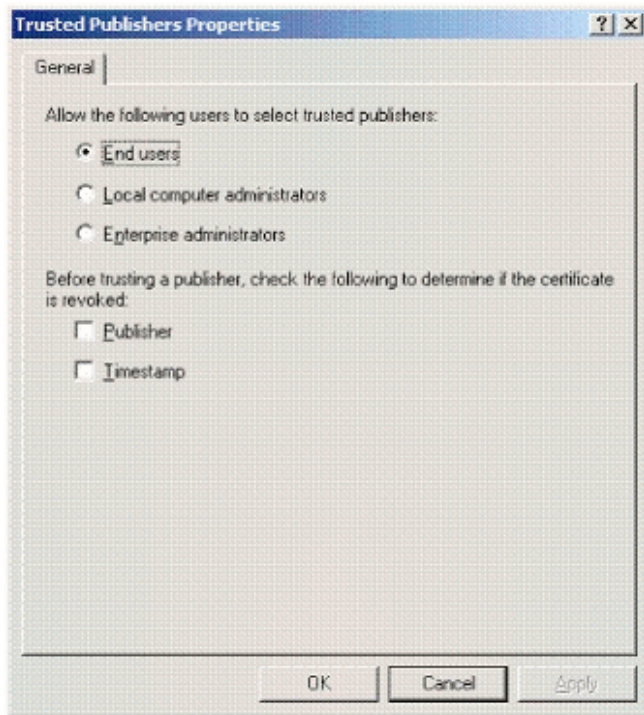


Figure 3: Designated File Types dialog box

[See full-sized image.](#)

### Trusted Publishers

The Trusted Publishers options shown in Figure 4 below allow you to configure settings related to ActiveX® controls and other signed content.



**Figure 4: Setting Trusted Publishers options**  
[See full-sized image.](#)

Table 3 shows Trusted Publisher options related to the use of ActiveX controls and other signed content.

**Table 3 Trusted Publisher Tasks and Settings**

Task	Setting
To allow only domain administrators to make decisions regarding signed active content	<b>Enterprise Administrators</b>
To allow local machine administrators to make all decisions regarding signed active content	<b>Local computer Administrators</b>
To allow any user to make decisions regarding signed active content	<b>End Users</b>
To ensure that the certificate used by the software publisher has not been revoked.	<b>Publisher</b>
To ensure that the certificate used by the organization that time-stamped the active content has not been revoked.	<b>Timestamp</b>

**Scope of Software Restriction Policies**

Software restriction policies do not apply to the following:

- Drivers or other kernel mode software.
- Any program run by the SYSTEM account.
- Macros inside of Microsoft Office 2000 or Office XP documents.
- Programs written for the common language runtime. (These programs use the Code Access Security Policy.)

[↑ Top of page](#)

## Software Restriction Policy Design

This section covers how software restriction policies are administered using Group Policy snap-ins, things to be concerned about when editing a policy for the first time, and what's involved in applying a software restriction policy to a group of users.

### Integration with Group Policy

Software restriction policies are administered using the following Group Policy snap-ins:

#### Domain Policy

To set up a domain policy

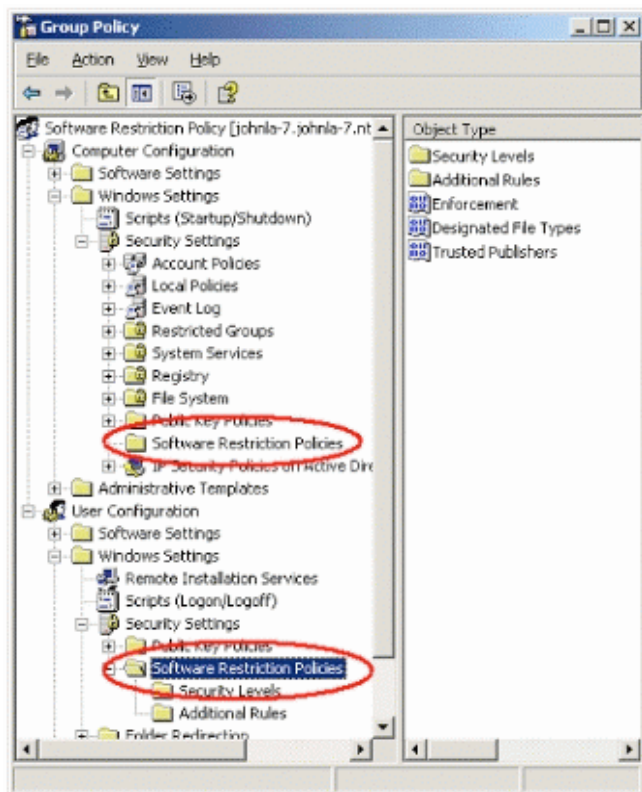
1. Click Start, then Run; type `dsa.msc` and click OK.
2. Right-click on domain or OU, then click Properties > Group Policy tab > New/Edit.

#### Local Security Policy

To set up a security policy

1. Click Start, then Run.
2. Type `secpol.msc`, then click **OK**.

If editing a GPO, you can set User and Machine software restriction policies as shown in Figure 5 below.



**Figure 5: Setting User and Machine software restriction policies**

[See full-sized image.](#)

If editing the local security policy, the software restriction policy settings are located as indicated in Figure 6 below.

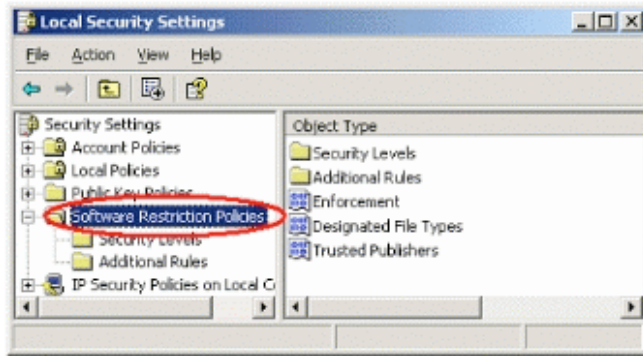


Figure 6: Editing Local Security Policy

[See full-sized image.](#)

### First-time Considerations

The first time you edit a policy you will see the message in Figure 7. The message is warning you that creating a policy will define default values. These default values can override settings from other software restriction policies.

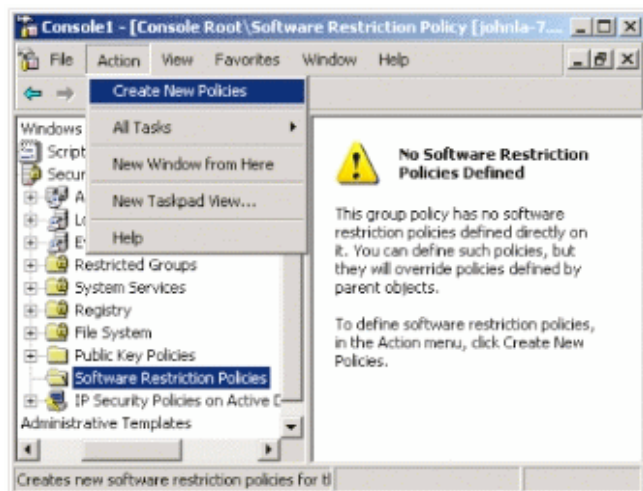


Figure 7: Warning message when creating a new policy

[See full-sized image.](#)

To create a policy:

- Select **Create New Policies** from the Action menu.

### Applying a Software Restriction Policy to a Group of Users

A software restriction policy is delivered through Group Policy to a site, domain, or organizational unit. However, an administrator may want to apply a software restriction policy to a group of users within a domain. To do this, the administrator can use GPO filtering.

For more information on GPO filtering see the article Windows 2000 Group Policy at <http://www.microsoft.com/windows2000/techinfo/howitworks/management/grouppolwp.asp>

### Terminal Servers

Software restriction policies are an integral part of securing a Windows Server 2003 terminal server. Terminal server administrators can now thoroughly lock down software access on a terminal server. Software restriction policies are even

more imperative on a terminal server because of the potentially vast number of users on a single machine. On a single-user Windows XP client, running a bad application inconveniences only one user, whereas running the same application on a terminal server could inconvenience more than 100 users. Software restriction policies prevent this problem. This service also removes the need for such applications as appsec.exe to govern software execution on a Windows Server 2003 terminal server.

In addition, Microsoft recommends that you view [278295](#) (How to Lock Down a Windows 2000 Terminal Server Session) to further lock down the client sessions on a terminal server.

Sometimes, several terminal servers have the same software installed on them, but their administrator wants to grant a certain group of users access to some software and a different group of users access to different software. Some software will be shared between the groups. For example, a law firm hosts its applications across a farm of terminal servers. The servers all have the same software installed. The access rules to the software are as follows:

- Any employee can use Microsoft Office and Internet Explorer. All employees are members of the *AllEmployees* group.
- Any accounting employee can use the Accounting Software. Accounting employees are members of the *AccountingEmployees* group.
- Any Lawyer can use the Law Research software. Lawyers are members of the *Lawyers* group.
- Any mailroom employee can use the Mail Room Processing software. Mailroom employees are members of the *MailRoomEmployees* group.
- Any executive can access all software available to all other employees. Executives are members of the *Executives* group.
- GPOs do not affect Administrators.

To achieve this software access, the administrator creates five Group Policy objects with customized software restriction policies. Each GPO is filtered so that only the users in the *AllEmployees*, *AccountingEmployees*, *Lawyers*, *MailRoomEmployees*, and *Executives* groups receive the GPO intended for them.

Because only executives should be able to access any software on their local workstations, as well as on the terminal servers, the administrator uses the loopback feature of Group Policy. The loopback feature allows an administrator to apply policy to a user based on the computer the user is logging onto. In loopback replace mode, the computer GPO settings are reapplied during user login, and the user GPO settings are ignored. See the Group Policy white paper for more information on how to configure loopback.

<b>User GPO: A1 Linked with Law Domain</b>	
<i>Filter: Law Domain Computers have Apply Group Policy permission</i>	
<b>Default Security Level</b>	
<i>Disallowed</i>	
<b>Path Rules</b>	
<i>%WINDIR%</i>	<i>Unrestricted</i>
<i>%PROGRAMFILES%\Common Files</i>	<i>Unrestricted</i>
<i>%PROGRAMFILES%\Internet Explorer</i>	<i>Unrestricted</i>
<i>%PROGRAMFILES%\Windows NT</i>	<i>Unrestricted</i>
<i>%PROGRAMFILES%\Microsoft Office</i>	<i>Unrestricted</i>

<b>User GPO: A2 Linked with Law Domain</b>	
--	--

<i>Filter: LawDomain Computers and AccountingEmployees have Apply Group Policy permission</i>	
<b>Default Security Level</b>	
<i>Disallowed</i>	
<b>Path Rules</b>	
<i>%PROGRAMFILES%\Accounting Software</i>	<i>Unrestricted</i>

<b>User GPO: A3 Linked with Law Domain</b>	
<i>Filter: Law Domain Computers and MailRoomEmployees have Apply Group Policy permission</i>	
<b>Default Security Level</b>	
<i>Disallowed</i>	
<b>Path Rules</b>	
<i>%PROGRAMFILES%\Mailroom Processing</i>	<i>Unrestricted</i>

<b>User GPO: A4 Linked with Law Domain</b>	
<i>Filter: Law Domain Computers and Lawyers have Apply Group Policy permission</i>	
<b>Default Security Level</b>	
<i>Disallowed</i>	
<b>Path Rules</b>	
<i>%PROGRAMFILES%\Law Research Software</i>	<i>Unrestricted</i>

<b>User GPO: A5 Linked with Lab Resource Domain</b>	
<i>Filter: Law Domain Computers and Executives have Apply Group Policy permission</i>	
Enable Loopback in Replace Mode	
<b>Default Security Level</b>	
<i>Disallowed</i>	
<b>Path Rules</b>	
<i>%PROGRAMFILES%\Law Research Software</i>	<i>Unrestricted</i>
<i>%PROGRAMFILES%\Mail Room Program</i>	<i>Unrestricted</i>
<i>%PROGRAMFILES%\Accounting Software</i>	<i>Unrestricted</i>

[↑ Top of page](#)

## Step-by-Step Guide for Designing a Software Restriction Policy

This section outlines the steps to follow when designing a software restriction policy.

### Items to Address

When designing a policy, decisions need to be made regarding the following items:

- GPO or local security policy
- User or machine policy
- Default security level
- Additional rules
- Policy options
- Linking the policy to a site, domain, or organizational unit

## Stepping Through the Process

### Step 1. GPO or Local Security Policy

Should the policy apply to many machines or users in a domain or organizational unit, or should it only apply to the local machine?

- If the policy should apply to many machines or users in a domain or other Active Directory container, use a GPO.
- If your policy should only apply to the local machine, use the Local Security Policy.

### Step 2. User or Machine Policy

Should the policy apply to users regardless of where they log in, or to a machine regardless of who logs in?

- If you want the policy to apply to a specific group of users, for example the Marketing Department domain group, then you need a user policy.
- If you want the policy to apply to a set of machines and all the users that log on to those machines, then you need a machine policy.

### Step 3. Default Security Level

Do you know all of the software your users will be running, or can they install any software they choose?

- If you know all of the software your users will be running, you should set the default security level to Disallowed.
- If users can install any software they want, set the default security level to Unrestricted.

### Step 4. Additional Rules

Identify the applications you choose to allow or disallow using the four rule types outlined in the Software Restriction Policy Architecture section above.

- To see which rules make sense for your policy, refer to Table 1. When to Use Each Rule, above.
- To create additional rules, refer to the Step-by-step Guide for Creating Additional Rules, below.

### Step 5. Policy Options

There are several policy options:

- If you are using a local security policy, and do not want the policy to apply to administrators on the machine, set the **Skip Administrators** option.
- If you want to check DLLs in addition to executables and scripts, turn on the **DLL checking** option.
- If you want to set rules on file types that are not in the default list of designated file types, then **add additional file types**.

- If you want to change who can make decisions about downloading ActiveX controls and other signed content, set **Trusted Publishers** options.

#### **Step 6. Linking the Policy to a Site, Domain, or Organizational Unit**

To link a GPO to a site.

1. Use the Active Directory **Sites and Services** snap-in.
2. Right-click the site, domain, or OU to which you want to link the GPO, and select **Properties**.
3. Select the **Group Policy** tab, to create, edit, and manage GPOs.

To link a GPO to a domain or OU,

1. Use the Active Directory **Users and Computers** snap-in.
2. Right-click the site, domain, or OU to which you want to link the GPO, and select **Properties**.
3. Select the **Group Policy** tab, to create, edit, and manage GPOs.

#### **Filtering**

GPO filtering can be done at this stage. You can have a portion of an OU receive a GPO by filtering based on group membership. You can also filter based on a WMI query.

#### **Testing A Policy**

If you want to test your policy immediately, instead of waiting for the next Group Policy refresh interval, run **gpupdate.exe** and log on again to test your policy.

[↑ Top of page](#)

### **Step-by-Step Guide for Creating Additional Rules**

The following steps are helpful when creating additional rules. To illustrate the principles behind the steps, each one illustrates an example of creating rules for Microsoft Office XP.

#### **Step 1. List the Software Applications**

List the software you are trying to identify. For our Office XP example, the software consists of Microsoft Word, Excel, PowerPoint®, and Outlook®.

#### **Step 2. Decide Rule Type**

Refer to Table 1. When to Use Each Rule, above, to decide which rule type to use. Also determine the security level for your rule. For our example, we use path rules set to the Unrestricted security level.

#### **Step 3. Record the Folders Where the Software is Installed**

List the paths where the software is installed. Three ways to do this include:

- You can look at the **Target** property of a shortcut to the file.
  - You can start each program by clicking Start, Run, and then typing msinfo32.exe. From msinfo32, select Software Environment and then Running Tasks.
  - You can use the following command: `wmic.exe process get "ExecutablePath, ProcessID"`

For our example, you will see the following tasks running:

- "C:\Program Files\Microsoft Office\Office10\WINWORD.EXE"
- "C:\Program Files\Microsoft Office\Office10\EXCEL.EXE"

- "C:\Program Files\Microsoft Office\Office10\POWERPNT.EXE"
- "C:\Program Files\Microsoft Office\Office10\OUTLOOK.EXE"

#### Step 4. Identify Dependent Programs

Some programs launch other programs to perform tasks. Your software application may depend on one or more supporting programs. For example, Microsoft Word launches the Microsoft Clip Organizer to manage clipart. The Microsoft Clip Organizer uses the following programs:

- C:\Program Files\Microsoft Office\Office10\MSTORDB.EXE
- C:\Program Files\Microsoft Office\Office10\MSTORE.EXE

Microsoft Office also uses files in the C:\Program Files\Common Files folder

#### Step 5. Generalize the Rules

In this step you should group related rules together to create a more general rule. Consider using environment variables, wild cards, and registry path rules.

Continuing our example, each program is stored in C:\Program Files\Microsoft Office\Office10, so it is sufficient to use one path rule for that folder instead of four separate path rules. Also, if Office is always installed in the Program Files folder on your machines, use an environment variable instead of an explicit path. Thus, our proposed rules are:

- %ProgramFiles%\Microsoft Office\Office10
- %ProgramFiles%\Common Files

#### Step 6. Have You Allowed Too Much?

This is the step where you look at what else is allowed by the rules you have proposed. Creating a rule that is too general may allow programs to run that you did not intend. The Office10 folder in our example also contains:

- FINDER.EXE
- OSA.EXE
- MCDLC.EXE
- WAVTOASF.EXE

Because these programs are acceptable to run, we do not have to change our rules.

[↑ Top of page](#)

## Commonly Overlooked Rules

When designing a policy, consider the following areas when creating rules.

### Login Scripts

Login scripts are stored on a central server. Often this central server can change with each login. If your default rule is Disallowed, be sure to create rules that identify the locations of your log on scripts. Consider using wildcards to identify these locations if the log on servers have similar names.

### System File Protection

System File Protection contains backup copies of many system programs in a folder named dllcache. These programs can be started by a user who knows the full path to the backup copy. If you want to disallow users running programs contained in the backup folder, you may want to create the following rule: **%WINDIR%\system32\dllcache, Disallowed**

### Common Startup Locations

Windows has many locations that contain links to programs that run at start up. If you don't make provisions for these programs, users will receive error messages when they log in.

Common startup locations include:

- %USERPROFILE%\Start Menu\Programs\Startup
- %ALLUSERSPROFILE%\Start Menu\Programs\Startup
- Win.ini, System.ini lines beginning with "run=" and "load="
- HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run
- HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

### Virus Scanning Programs

Most anti-virus software has a real-time scanner program that starts when the user logs in and scans all files accessed by the user, looking for possible virus contamination. Make sure your rules allow your virus scanning programs to run.

[↑ Top of page](#)

## Scenarios

This section examines some typical problems and how software restriction policies can be used to solve them.

### Block Malicious Scripts

An organization wants to be protected from script-based viruses. The LoveLetter virus, technically called a worm, was estimated to have caused between \$6 and \$10 billion in damage. This worm, which has more than 80 variants, continue to be encountered frequently.

The LoveLetter worm, written in the Visual Basic Script language (VBS), is encountered as LOVE-LETTER-FOR-YOU.TXT.VBS. A software restriction policy blocks this worm simply by disallowing any .vbs file from running.

However, many organizations use VBS files for systems management and logon scripts. Blocking all VBS files from running protects an organization, but a VBS can no longer be used for legitimate purposes. A software restriction policy overcomes this handicap by blocking the undesirable VBS, while allowing legitimate ones to run.

This policy can be created using the rules in Table 4.

**Table 4 Rules for Blocking Malicious Scripts**

Default Security Level: Unrestricted	
Path Rules	
*.VBS	Disallowed
*.VBE	Disallowed
*.JS	Disallowed
*.JSE	Disallowed
*.WSF	Disallowed
*.WSH	Disallowed
Certificate Rules	
IT Department Certificate	Unrestricted

This policy prevents all scripting files associated with the Windows Scripting Host from running, except those that are digitally signed by the IT Department certificate. See Appendix below for how to obtain a certificate and digitally sign files.

### Manage Software Installation

You can configure your organization's machines so that only approved software can be installed. For software that uses Windows Installer technology, this can be accomplished by the policy shown in Table 5.

**Table 5 Rules for Managing Software Installation**

Default Security Level: Unrestricted	
Path Rules	
*.MSI	Disallowed
\\products\install\PROPLUS.MSI	Unrestricted
Certificate Rules	
IT Department Certificate	Unrestricted

This policy prevents all Windows Installer packages from installing. It allows MSI files digitally signed by the IT department certificate and the OWC10.MSI package located at \\products\install to be installed. See the Appendix below for how to obtain a certificate and digitally sign files.

This policy also shows how you can use the precedence of the path and certificate rules to allow just the software you want. For any other package that your organization cannot or does not want to digitally sign, you can create hash rules, or fully qualified path rules, to make exceptions for them.

### Line-of-Business PC

In some cases an administrator may want to manage all of the software that runs on a machine. This is because even when users have insufficient rights to replace system files or files in shared folders such as Program Files, if they have a place on the file system they can write to, then they can also copy a program there and start it up.

Viruses contracted this way can damage the system by modifying operating system settings and files; they can also cause great damage by misusing the user's privileges. For example, mass-mailer worms can be spread by accessing the user's address book and sending mail. Even normal users on a system are vulnerable to this kind of attack.

As long as users are not administrators on their local machines, the policy in Table 6 protects them from accidentally running malicious code. Because users cannot modify the contents of the Program Files or Windows folders, they can only run software installed by an administrator.

**Table 6 Policy for Managing all Software on a Machine**

Default Security Level: Disallowed	
Apply software restriction policies to the following users:	
All users except administrators	
Path Rules	
%WINDIR%	Unrestricted
%PROGRAMFILES%	Unrestricted

This policy disallows all software on the user's machine, except that installed in the Windows directory, Program Files

directory, or their respective subfolders. It does not apply to administrators.

If a user receives a virus attachment in an e-mail, for example WORM.vbs, the mail program will copy it to the profile directory (%USERPROFILE%) and launch it from there. Because the profile directory is not a subfolder of the Windows folder or the Program Files folder, programs launched from there will not run.

If all the programs a user needs are not installed in %WINDIR% or %PROGRAMFILES%, or there are programs in those folders that the administrator does not want the user running, the administrator can make additional exceptions as show in Table 7.

**Table 7 Exceptions for Managing all Software on a Machine**

Path Rules	
%WINDIR%\regedit.exe	Disallowed
%WINDIR%\system32\cmd.exe	Disallowed
\\CORP_DC_??\scripts	Unrestricted
%HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates \InoculateIT\6.0\Path\HOME%	Unrestricted

The effects of these exceptions are:

- Both the command prompt (cmd.exe) and the registry editor (regedit.exe) are disallowed.
- An exception is created to allow login scripts to run on the user's machine.
- The use of the "?" wildcard allows the rule to match \\CORP\_DC\_01, \\CORP\_DC\_02, and others.
- A registry path rule is added that allows the anti-virus software on the machine to run.

### Different Policies for Different Users

In this scenario, there are machines that are shared by many users. The machines have the same software installed on them, but the administrator wants to grant a certain group of users access to some software, and a different group of users access to other software. There also will be software that is shared between the groups.

#### Example

A computer lab at a university runs 15 machines with identical software. They have Microsoft Office, computer-aided design (CAD) software, and the Microsoft Visual C++® compiler. For licensing reasons, the administrators of the computer lab want to ensure the following:

- Any student can use Microsoft Office—all students are members of the AllStudents group.
- Any engineering student can use the CAD software—engineering students are members of the EngStudents group.
- Any computer science student can use the Microsoft Visual C++ compiler—computer science students are members of the CSStudents group.

To achieve the objectives of the above scenario, the administrator creates three Group Policy objects with customized software restriction policies. Each GPO is filtered so that only the users in AllStudents, EngStudents, and CSStudents receive the GPO intended for them.

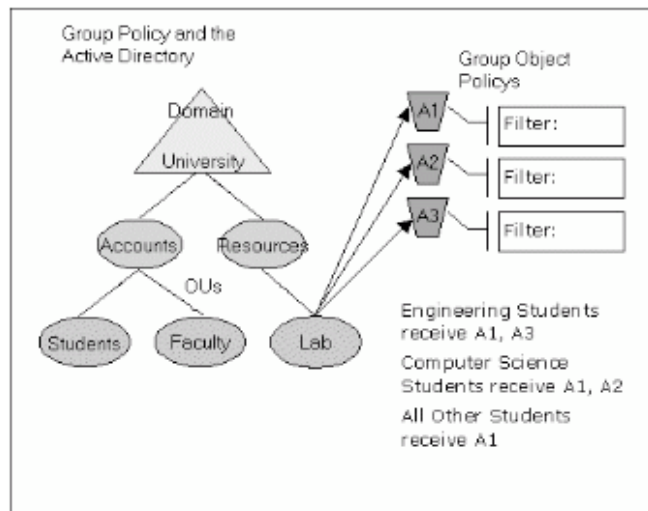
Because the administrator wants the students to receive the policy when logged on to the lab computers, but not when the students log on to their personal computers, he uses the Group Policy loopback feature. The loopback feature allows an administrator to apply policy to a user based on the computer the user is logging on to. In loopback replace mode, the machine GPOs are reapplied during user logon, skipping the normal user policies.

Refer to Tables 8, 9 and 10, and Figure 8 below.

For more information on how to configure loopback, see the article Windows 2000 Group Policy at <http://www.microsoft.com/windows2000/techinfo/howitworks/management/groupplwp.asp>

**Table 8 A1 Linked with Lab Resource Domain**

User GPO: A1 Linked with Lab Resource Domain	
Filter: Domain Computers have Apply Group Policy permission	
Default Security Level	
Disallowed	
Path Rules	
%WINDIR%	Unrestricted
%PROGRAMFILES%\Common Files	Unrestricted
%PROGRAMFILES%\Messenger	Unrestricted
%PROGRAMFILES%\Internet Explorer	Unrestricted
%PROGRAMFILES%\Windows Media Player	Unrestricted
%PROGRAMFILES%\Windows NT	Unrestricted



**Figure 8: Group Policy Organization for Computer Lab**

[See full-sized image.](#)

**Table 9 A2 Linked with Lab Resource Domain**

User GPO: A2 Linked with Lab Resource Domain	
Filter: Domain Computers and CSStudents have Apply Group Policy permission	
Enable Loopback in Replace Mode	
Default Security Level	
Disallowed	

**Table 9 A2 Linked with Lab Resource Domain**

Path Rules	
%PROGRAMFILES%\Microsoft Visual Studio	Unrestricted

**Table 10 A3 Linked with Lab Resource Domain**

User GPO: A3 Linked with Lab Resource Domain	
Filter: Domain Computers and EngStudents have Apply Group Policy permission	
Enable Loopback in Replace Mode	
Default Security Level	
Disallowed	
Path Rules	
%PROGRAMFILES%\CAD Application	Unrestricted

[↑ Top of page](#)

## Deployment Considerations

This section covers a variety of issues that need to be considered when deploying software restriction policies.

### Best Practices

Best practices to be followed when deploying software restriction policies include:

**Always create a separate GPO for software restriction policies.** If you create a separate GPO for your policy settings, you can disable it in an emergency without affecting the rest of your security settings.

**Never modify the default domain policy.** If you do not edit the default policy, you always have the option of reapplying it

**Never link to a software restriction policy in another domain.** Linking to a Group Policy object in another domain can result in poor performance.

**Thoroughly test new policy settings in test environments before applying the policy settings to your domain.** New policy settings might act differently than you originally expected. Testing diminishes the chance of encountering a problem when you deploy policy settings across your network.

- You can set up a test domain, separate from your organization's domain, in which to test new policy settings.
- You can also test the policy settings by creating a test GPO and linking it to an OU. When you have thoroughly tested the policy settings with test users, you can link the test GPO to your domain.
- Typing mistakes, or incorrectly entered information, can result in a policy setting that does not perform as expected. Testing new policy settings before applying them can prevent unexpected behavior.
- Do not disallow programs or files without testing to see what the effect might be. Restrictions on certain files can seriously affect the operation of your computer or network.

### Group Policy Processing

The following information needs to be considered when working with Group Policy objects:

**Use security groups to filter the scope of the Group Policy object.** You can further refine which groups of computers and users a particular GPO influences by using Windows 2000 security groups.

- Use the Security property page of a given GPO to set access permissions (Discretionary Access Control Lists or DACLs) to allow or deny access to the GPO by specified groups.

For more information on GPO filtering, see the article Windows 2000 Group Policy at <http://www.microsoft.com/windows2000/techinfo/howitworks/management/grouppolwp.asp>

**Use WMI to filter the scope of the Group Policy object.** Windows XP clients support WMI filtering of GPOs. This allows a client to skip processing a GPO based on WMI information available on the client.

- Use the WMI Filter property page of a given GPO to add a WMI filter. For example, you can create a WMI filter so that a GPO only applies to machines with a certain service pack.

For more information on WMI filtering, see the article Windows 2000 Group Policy at <http://www.microsoft.com/windows2000/techinfo/howitworks/management/grouppolwp.asp>

**Order of Group Policy application.** By default, Group Policy is inherited and cumulative, and it affects all computers and users in an Active Directory container. Group Policy objects are processed according to the following order:

- Local GPO is applied
- GPOs linked to sites
- GPOs linked to domain
- GPOs linked to OUs. (In the case of nested OUs, GPOs associated with parent OUs are processed prior to GPOs associated with child OUs.)

This order of GPO processing (local ? site ? domain ? OU) is significant because policy applied later overwrites policy applied earlier.

**No Override and Block Policy Inheritance Options.** You can enforce the Group Policy settings in a specific Group Policy object by using the No Override option so that GPOs in lower-level Active Directory containers are prevented from overriding that policy.

You can also block inheritance of Group Policy from parent Active Directory containers by using the Block policy inheritance option.

See the article Windows 2000 Group Policy—<http://www.microsoft.com/windows2000/techinfo/howitworks/management/grouppolwp.asp> for more information.

### Mixed Domain Deployments

It is possible to use software restriction policies in a mixed-mode deployment. That is, you do not have to upgrade your Windows 2000 domain controllers to take advantage of software restriction policies. You can use a Windows XP Professional computer to edit the Group Policy object and configure your software restriction policy. Windows XP and Windows Server 2003 computers that download the GPO will enforce the software restriction policy. Computers running Windows 2000 will ignore the settings.

### Merging Semantics for Multiple Software Restriction Policies

Whenever two or more Group Policy objects apply to a user or machine, the policies are merged. When two or more software restriction policies are merged, the following occurs:

- The GPO with the highest precedence sets the following values:
  - Default Security Level
  - Designated File Types
  - Skip Administrators
  - DLL Checking

- The rules from multiple GPOs are always merged. Thus, all additional rules from all GPOs are preserved.

A software restriction policy can be set for user scope and machine scope. The following semantics are observed when merging user and machine scope:

- The more restrictive default security level is chosen.
- The list of designated file types in the machine policy, if present, is used. If not present, the list of designated file types in the user policy is used.
- The Skip Administrators value is always chosen from the machine policy.
- If DLL checking is enabled in either policy, then it is enabled.
- All the rules between user and machine policies are merged.

[↑ Top of page](#)

## Troubleshooting Software Restriction Policies

This section includes information for troubleshooting problems with software restriction policies.

### Default Settings for a Software Restriction Policy

The default settings for a software restriction policy include the following:

- Default Security Level: Unrestricted
- Enforcement options:
  - Apply to Files: All software files except libraries (such as DLLs)
  - Apply to Users: All users
- Additional Rules: none
- Designated File Types: See Table 11. Default Designated File Types in the Appendix below.
- Trusted Publishers:
  - Select Trusted Publishers: End Users
  - Publisher Certificate Revocation Checking: Not selected
  - Timestamp Certificate Revocation Checking: Not selected

### Error Message

When a program is disallowed due to a software restriction policy, an error code is received by the launching program. If the launching program returns the system message for this error code, you will see the following message:

"Windows cannot open this program because it has been prevented by a software restriction policy. For more information, open Event Viewer or contact your system administrator," as shown in Figure 9 below.

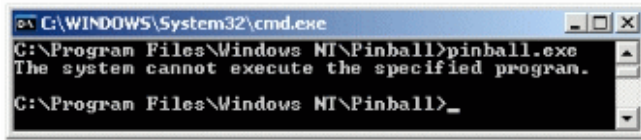


**Figure 9:** Error message received in the Windows Explorer

[See full-sized image.](#)

Some programs display one error message for many possible error codes. For example, the Windows command prompt displays the following message when a program is restricted by a software restriction policy:

"The system cannot execute the specified program," as shown in Figure 10 below.



**Figure 10: Error message received in command prompt**

[See full-sized image.](#)

## Rule GUIDs

Each path, hash, or zone rule has an associated globally unique identifier (GUID). Even two identical rules, for example two disallowed hash rules on the same program, have a different GUID associated with each. This GUID is stored in the registry along with the rule data. Various logging and troubleshooting tools reveal these GUIDs. A rule GUID enables you to determine GPOs where a rule is defined.

### The Case of the Missing Calculator

To see how the GUID can aid troubleshooting, consider an example where a user attempts to start the program called `calc.exe`, the Windows calculator. The user receives the error that it has been prevented by a software restriction policy. Thinking this a mistake, the user places a call into the help desk call center. The support professional checks the event log and sees the following software restriction policy event.

- Access to `C:\WINDOWS\system32\calc.exe` has been restricted by your Administrator by location with policy rule `{91ecff50-2ff4-4672-a182-b0f07a74b2df}` placed on path `C:\WINDOWS\system32\calc.exe`

The event log entry detail shows the GUID `{91ecff50-2ff4-4672-a182-b0f07a74b2df}`. The support professional runs the `gresult.exe` tool and sees the following entry:

- GPO: DisallowedPolicy
- Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\Paths\{91ecff50-2ff4-4672-a182-b0f07a74b2df}
- State: Enabled

The support professional opens up the GPO named `DisallowedPolicy` in the Group Policy editor. Examining the rules, the support professional sees a path rule for `calc.exe`. The description in the rule indicates that it is supposed to disable the program `cacls.exe`, which is used to display or modify access control lists (ACLs) of files. The support professional concludes that a typo was made in the rule, where `calc.exe` was entered instead of `cacls.exe`, and follows up with the appropriate IT administrator.

## Event Log

Software restriction policies can generate the following event log entries:

```
Event Log: System
Event Type: Warning
Event Source: Software Restriction Policy
Event Category: None
Event ID: 865
Date: 6/6/2001
Time: 2:50:29 PM
User: bob
Computer: EXAIR-1
Description:
Access to C:\Program Files\Messenger\msmsgs.exe has been restricted by your
Administrator by the default software restriction policy level.
This event is logged when a user starts a program that is disallowed by the default
```

```

security level.
Event Log:      System
Event Type:     Warning
Event Source:   Software Restriction Policy
Event Category: None
Event ID:      866
Date:          6/6/2001
Time:          2:50:29 PM
User:          bob
Computer:      EXAIR-1
Description:
Access to C:\Program Files\Messenger\msmsgs.exe has been restricted by your
Administrator by location with policy rule {79d2f45e-5d93-4138-9608-dde4afc8ac64}
placed on path C:\Program Files\Messenger\msmsgs.exe
This event is logged when a user starts a program that is disallowed by a path rule.
The rule GUID, {79d2f45e-5d93-4138-9608-dde4afc8ac64} in this example, can be used in
conjunction with gpresult.exe to find the GPO this rule came from.
Event Log:      System
Event Type:     Warning
Event Source:   Software Restriction Policy
Event Category: None
Event ID:      867
Date:          6/6/2001
Time:          2:50:29 PM
User:          bob
Computer:      EXAIR-1
Description:
Access to C:\Program Files\Messenger\msmsgs.exe has been restricted by your
Administrator by software publisher policy.
This event is logged when a user starts a program that is disallowed by a
certificate rule.
Event Log:      System
Event Type:     Warning
Event Source:   Software Restriction Policy
Event Category: None
Event ID:      868
Date:          6/6/2001
Time:          2:50:29 PM
User:          bob
Computer:      EXAIR-1
Description:
Access to C:\Program Files\Messenger\msmsgs.exe has been restricted by your
Administrator by policy rule {79d2f45e-5d93-4138-9608-dde4afc8ac64}.
This event is logged when a user starts a program that is disallowed by a zone rule or
hash rule.

```

The following command line will query for all software restriction policy events. To refine the query, consult the usage of EventQuery by typing "EventQuery /?" at the command line.

```
EventQuery -l System -fi "ID ge 865" -fi "ID le 868" -v -fo list
```

### Advanced Logging

When creating rules or troubleshooting a machine displaying problems, an administrator may want a log of every software restriction policy evaluation. This can be done by enabling advanced logging.

To enable advanced logging:

- Create the following registry key:

```
KEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers
```

```
String Value: LogFileName, <path to a log file>
```

### Enabling and Disabling Logging From the Command Line

The following commands can be used to enable and disable logging from the command line.

- Enable logging:

```
reg.exe add  
"HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers" /v LogFileName /d saferlog.txt
```

- Disable logging:

```
reg.exe delete  
"HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers" /v LogFileName /f
```

- A log file entry is formatted as follows:

parent process (Process ID) identified Path to launched program as Rule Level using Rule Type, GUID = GUID of rule

- Example entry:

```
winlogon.exe (PID = 396) identified C:\Windows\system32\userinit.exe as Unrestricted using path rule, Guid =  
{f8c2c158-e1af-4695-bc93-07cbefbdc594}
```

This entry shows that the winlogon process, running with process ID 396, launched the program C:\Windows\system32\userinit.exe. The rule that matched the program had a GUID of {f8c2c158-e1af-4695-bc93-07cbefbdc594}. The security level for this rule was Unrestricted. The matching rule was a path rule.

**Note:** When not performing advanced logging be sure to turn it off by deleting the registry value. Using advanced logging over a long period of time can consume a large amount of disk space and slow system performance.

## Group Policy Troubleshooting

The following tools are used to troubleshoot Group Policy problems.

### Resultant Set of Policy (RSOP)

RSOP is an infrastructure and tool in the form of MMC snap-ins, enabling administrators to determine and analyze the current set of policies in two modes: logging mode and planning mode. In logging mode, administrators assess what has been applied to a particular target. In planning mode, administrators can see how policies would be applied to a target, and then examine the results before deploying a change to Group Policy.

To view RSOP data for the current user

- Click **Start, Run**, and type **rsop.msc**

### gpupdate.exe

Gpupdate is a utility for Group Policy. It can cause a refresh of Group Policy on the client machine and can be used for software restriction policies in the following ways:

- **gpupdate /target:Computer [/Force]** This command refreshes the machine-based software restriction policy settings. The /Force switch, if present, instructs the machine to reapply all settings, regardless of whether they have changed since the last Group Policy refresh.
- **gpupdate /target:User [/Force]** This command refreshes the user-based software restriction policy settings. The /Force switch, if present, instructs the machine to reapply all settings, regardless of whether they have changed since last Group Policy refresh.
- **gpupdate [/Force]** This command refreshes the user- and machine-based software restriction policy settings. The /Force switch, if present, instructs the machine to reapply all settings, regardless of whether they have changed since the last Group Policy refresh.

After refreshing software restriction policy settings, only new programs started will enforce the policy. Some long-lived programs like explorer.exe, the Windows shell, will not pick up the new policy. To force all programs to enforce the policy, the user should log in again.

## gpresult.exe

Gpresult.exe is a Group Policy utility for examining the settings applied during Group Policy refresh. It utilizes Resultant Set of Policy (RSOP) data. It can be used for software restriction policies in the following ways:

- **gpresult.** This command displays basic user and machine information. It lists the group policies that apply to the logged in user on the current machine.

## Command Sample

The following is sample output from the command: **gpresult /scope user /v /user bob.**

```
Microsoft® Windows® XP Operating System Group Policy Result tool v2.0
Copyright© Microsoft Corp. 1981-2001
Created On 8/1/2001 at 3:07:34 PM
RSOP results for EXAIR-70\bob on EXAIR-7 : Logging Mode
OS Type: Microsoft Windows XP Server
OS Configuration: Primary Domain Controller
OS Version: 5.1.3524
Domain Name: EXAIR-70
Domain Type: Windows 2000
Site Name: Default-First-Site-Name
Roaming Profile:
Local Profile: C:\Documents and Settings\bob
Connected over a slow link?: No
User Settings
CN=bob,OU=Product Group,DC=EXAIR-7,DC=nttest,DC=microsoft,DC=com
Last time Group Policy was applied: 8/1/2001 at 2:49:28 PM
Group Policy was applied from: N/A
Group Policy slow link threshold: 500 kbps
Applied Group Policy Objects
  DisallowedPolicy
  Software Restriction Policy
  Default Domain Policy
```

**Note:** The Applied Group Policy Objects shows you the GPOs that are applied for this user.

```
The following GPOs were not applied because they were filtered out:
Local Group Policy
Filtering: Not Applied (Empty)
The user is a part of the following security groups:
  Domain Users
  Everyone
  BUILTIN\Users
  BUILTIN\Pre-Windows 2000 Compatible Access
  LOCAL
  NT AUTHORITY\INTERACTIVE
  NT AUTHORITY\Authenticated Users
```

**Note:** The group membership is listed here for troubleshooting GPO filtering scenarios.

```
Resultant Set Of Policies for User:
Software Installations: N/A
Public Key Policies: N/A
Administrative Templates
GPO: Software Restriction Policy
Setting:
Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\Paths\
{593905cd-1a5b-4c56-93a6-ecf1c8a78c04}
State: Enabled
```

**Note:** The rule detail is not displayed, but the GUID corresponding to the rule is displayed. The name of the GPO the setting comes from is also displayed.

```

GPO: DisallowedPolicy
Setting:
Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\Paths\{094a935d-a2b8-48be-a50b-0fe3174e9ced}
State: Enabled
GPO: DisallowedPolicy
Setting:
Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\Paths\{bba39f11-e1a9-406a-8296-3b2cbcb1f144}
State: Enabled
GPO: Software Restriction Policy
Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\Paths\{c0193a34-594d-452b-b3e6-edc0d593f345}
State: Enabled
GPO: DisallowedPolicy
Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers
State: Enabled
GPO: Software Restriction Policy
Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\Paths\{a5c5639e-4ee7-4882-aa80-560bbeccaca22}
State: Enabled
GPO: Software Restriction Policy
Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\Paths\{f63296b7-4b0a-4318-ae8d-5d070b44b4ec}
State: Enabled
GPO: DisallowedPolicy
Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers
State: Enabled
GPO: Software Restriction Policy
Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\Paths\{8e85c506-2964-4745-8f4e-3c2efe02f509}
State: Enabled
GPO: Software Restriction Policy
Setting: Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\Paths\{739c2db8-8ef5-4b2d-b210-d84d7b697603}
State: Enabled
Folder Redirection: N/A
Internet Explorer Browser User Interface: N/A
Internet Explorer Connection: N/A
Internet Explorer URLs: N/A
Internet Explorer Security: N/A
Internet Explorer Programs: N/A

```

## Recovery Options

When you start Windows in safe mode and log in as local administrator, the software restriction policy is not applied. Safe mode will let you fix a policy that is causing problems.

To fix a policy that is causing problems

1. Use the Group Policy snap-in to fix the policy.
2. Run **gpupdate.exe**.
3. Restart Windows and log in normally.

## Appendix

This section includes a list of default designated file types, registry formats and a how-to guide for digitally signing files with test certificates,

**Table 11 Default Designated File Types**

File Extension	File Description
. ADE	Microsoft Access Project Extension

**Table 11 Default Designated File Types**

<b>File Extension</b>	<b>File Description</b>
.ADP	Microsoft Access Project
.BAS	Visual Basic® Class Module
.BAT	Batch File
.CHM	Compiled HTML Help File
.CMD	Windows NT® Command Script
.COM	MS-DOS® Application
.CPL	Control Panel Extension
.CRT	Security Certificate
.EXE	Application
.HLP	Windows Help File
.HTA	HTML Applications
.INF	Setup Information File
.INS	Internet Communication Settings
.ISP	Internet Communication Settings
.JS	JScript® File
.JSE	JScript Encoded Script File
.LNK	Shortcut
.MDB	Microsoft Access Application
.MDE	Microsoft Access MDE Database
.MSC	Microsoft Common Console Document
.MSI	Windows Installer Package
.MSP	Windows Installer Patch
.MST	Visual Test Source File
.PCD	Photo CD Image
.PIF	Shortcut to MS-DOS Program
.REG	Registration Entries
.SCR	Screen Saver
.SCT	Windows Script Component
.SHS	Shell Scrap Object
.URL	Internet Shortcut (Uniform Resource Locator)
.VB	VBScript File

**Table 11 Default Designated File Types**

File Extension	File Description
. VBE	VBScript Encoded Script File
. VBS	VBScript Script File
. WSC	Windows Script Component
. WSF	Windows Script File
. WSH	Windows Scripting Host Settings File

### Registry Format

After a policy is applied, the software restriction policy configuration is stored in the system registry. The security access control list (ACL) protecting these registry keys allows only administrators and the SYSTEM account to change them.

#### User Policy

User policy is stored under the following key:

HKEY\_CURRENT\_USER\Software\Policies\Microsoft\Windows\

#### Machine Policy


Machine policy is stored under the following key:


HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Microsoft\Windows\


#### Registry Format Explained

 [HKCU or HKLM]\SOFTWARE\Policies\Microsoft\Windows\Safer

 CodeIdentifiers


 DefaultLevel, DWORD (40000 for Unrestricted, 0 for Disallowed)

 ExecutableTypes, REG\_MULTI\_SZ (list of extensions for Designated File Types)

 TransparentEnabled, DWORD, (0 for No Enforcement, 1 for Skip DLLs, 2 for all files)

 PolicyScope, DWORD, (0 for All Users, 1 for Skip Administrators) HKLM only

[Optional registry values. These must be set manually]


 AuthenticcodeEnabled, DWORD, (1 for Apply Certificate Rules to EXE's) HKLM only

 LogFileName, REG\_SZ (Path to log file, turns on advanced logging) HKLM only

 0


**Note:** Entries under this key are Disallowed **rules**

 Hashes

 {0140090a-6e4d-4dc3-b1fa-27563cc91fda}


**Note:** Each number in braces is a GUID. Each GUID is unique.


 Description, REG\_SZ (text description)

 FriendlyName, REG\_SZ (File version information)

 ItemData, REG\_BINARY, (Hash value)

ItemSize, QWORD, (Size of the file)


 HashAlg, DWORD, (32771 is MD5, 32772 is SHA1)

 LastModified, QWORD, (Timestamp)

 SaferFlags, DWORD, (not used)

 Path

 {5c03dc31-e128-426e-bad6-9223ee92d0b8}


 Description, REG\_SZ (text description)

 ItemData, REG\_SZ (Path entry)


or


 ItemData, REG\_EXPAND\_SZ

**Note:** REG\_EXPAND\_SZ is used with path rules using environment variables and registry path rules


 LastModified, QWORD, (Timestamp)

 SaferFlags, DWORD, (not used)


 UrlZones

 {dda3f824-d8cb-441b-834d-be2efd2c1a33}


 ItemData, DWORD (Identifier for zone)


 LastModified, QWORD, (Timestamp)

 SaferFlags, DWORD, (not used)


 262144

**Note:** Entries under this key are Unrestricted Rules

 Hashes

 {0140090a-6e4d-4dc3-b1fa-27563cc91fda}

 Paths

 {302fe78d-0b85-484a-b16f-0ae6262b7969}

### Certificate Rules

Certificate rules are stored in a separate key in the registry.

Certificate rules for user software restriction policies are stored in this registry key:

- HKEY\_CURRENT\_USER\SOFTWARE\Policies\Microsoft\SystemCertificates


Certificate rules for machine software restriction policies are stored in this registry key:


- HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Microsoft\SystemCertificates

 [HKCU or HKLM]\SOFTWARE\Policies\Microsoft\Windows\SystemCertificates


 TrustedPublishers

**Note:** Entries under this key are Unrestricted rules

 Certificates

 D4C408A1F8EF6B49F837C54E5F697DC11EEB3F53

**Note:** This is a hash of the certificate

 Blob, REG\_BINARY (binary value of certificate)


 Disallowed

**Note:** Entries under this key are Disallowed rules

 Certificates

 C9902A94036312086FFAD974760D96CA93284555


**Note:** This is a hash of the certificate


 Blob, REG\_BINARY (binary value of certificate)


#### Default Settings


 [HKCU or HKLM]\SOFTWARE\Policies\Microsoft\Windows\Safer

 CodeIdentifiers

 DefaultLevel, DWORD (40000)


 ExecutableTypes, REG\_MULTI\_SZ (WSC,VB,URL,SHS, SCR, REG,PIF,PCD, OCX, MST,MSP, MSI, MSC, MDE,MDB,LNK, ISP,INS,INF,HTA,HLP,EXE, CRT, CPL,COM,CMD,CHM, BAT,BAS,ADP,ADE)

 TransparentEnabled, DWORD, (1 for Skip DLLs)

 PolicyScope, DWORD, (0)

 0


 Path


 {dda3f824-d8cb-441b-834d-be2efd2c1a33}

 Description, REG\_SZ

 ItemData, REG\_EXPAND\_SZ

%HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Cache%OLK\*

 LastModified, QWORD, (Timestamp)

 SaferFlags, DWORD, (0)

## Step-by-Step Guide to Digitally Signing Files with Test Certificates

This section examines tools and processes for digitally signing files used with certificate rules.

### Step 1: Download the Tools

Download the Authenticode for Internet Explorer 5.0. These tools are used to sign and verify files using Authenticode signatures.

<http://msdn.microsoft.com/downloads/default.aspx>

### Step 2: Enrolling for a Code-signing Certificate

The next step is to obtain a certificate that's valid for code-signing. There are three ways to do this:

- **Enroll for a code-signing certificate from a commercial certificate authority** such as VeriSign. If you want the digital signatures of your files to be valid outside of your organization, you should choose this option.
- **Set up a Windows 2000 or Windows Server 2003 certificate authority.** Enroll for a certificate against this CA. If only people in your organization use your digitally signed files, you should choose this option.
- **Create a self-signed certificate for test purposes.** After downloading the Authenticode tools, run the following two commands:
  - `makecert.exe -n "cn=TEST CERTIFICATE (FOR TEST PURPOSES ONLY!)" -ss my -eku 1.3.6.1.5.5.7.3.3`
  - `Setreg.exe 1 true`

The `setreg.exe` command instructs the local computer to trust the Test Root Agency certificate that issues your test code-signing certificate. You should not trust the test root certificate on production machines.

### Step 3: Signing a File

Create a test VB Script file called `hello.vbs` with the following contents:

- `msgbox "hello world"`

Sign and timestamp this file by running the following command:

- `signcode.exe -cn "TEST CERTIFICATE (FOR TEST PURPOSES ONLY!)" -t http://timestamp.verisign.com/scripts/timestamp.dll hello.vbs`

If the signing and time stamping operation is successful, the tool will print "Succeeded" at its completion. The script will have a Base 64 encoded digital signature section added to it as shown in Figure 11 below.

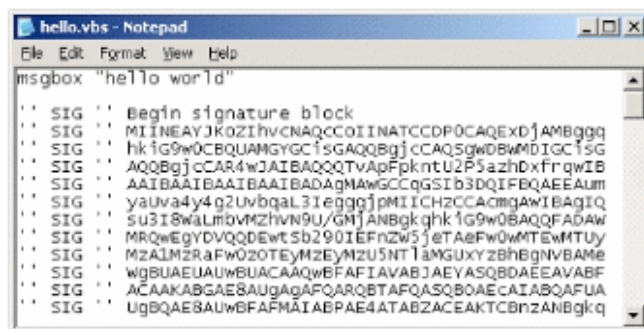


Figure 11: Visual Basic Script file with a digital signature

You can verify that the file was signed properly by running the following command:

- `chktrust.exe hello.vbs`

The dialog box in Figure 12 will appear.

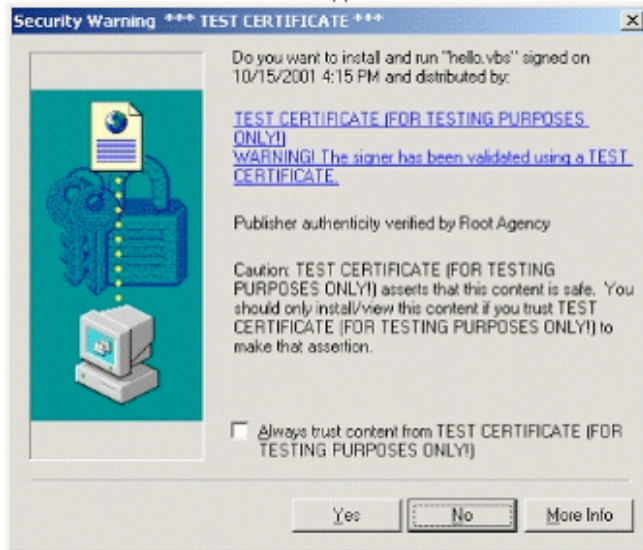


Figure 12: Verifying a signed file

#### Step 4: Create Certificate and Path Rules

Edit the local security policy—secpol.msc. Create two rules:

- New Path Rule: Type "\*.VBS" in the edit box labeled Path. Set the security level to Disallowed
- New Certificate Rule: Create a certificate rule for your test publisher certificate with a security level set to Unrestricted

Run the following command to export the certificate to a file. Browse to this file when creating the certificate rule.

- `certmgr.exe -put -c -v -n "TEST CERTIFICATE (FOR TESTING PURPOSES ONLY!)" -s my mytestcert.cer`

These two rules combine to disallow any VB Script file; except those that are signed by this test certificate.

Your policy should look like the example shown in Figure 13 below.

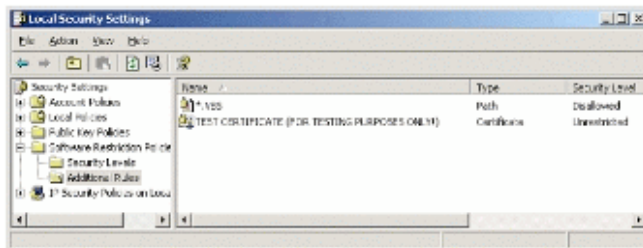


Figure 13: Software restriction policy showing certificate and path rules

#### Step 5: Re-login and Test the Software Restriction Policy

1. Log off, then log back on to ensure your machine is using the new rules.
2. Run hello.vbs. You should see a message box appear. This indicates the script was signed by the appropriate certificate and complied to the rules of the software restriction policy.

3. Edit hello.vbs with notepad and change the script to print **"Hello world. This script has been changed."** Save it, leaving the digital signature portion of the script intact.
4. Run the script again. You will notice that it is prevented from running because the digital signature on the script no longer verifies.

## Summary

Software restriction policies provide administrators with a policy-driven mechanism to identify software running on computers in a domain, and control its ability to execute. Policies can be used to block malicious scripts, help lockdown a computer, or prevent unwanted applications from running. They can be used in standalone mode or managed through Group Policy, and can be tailored to meet the needs of a set of users or computers. Software restriction policies promote improved system integrity and manageability—and ultimately lower the cost of owning a computer.

## Related Links

See the following resources for further information:

- [Technical Overview of Security Services](#)
- [Technical Overview of Terminal Services](#)
- [Windows 2000 Group Policy](#)
- [Whats New in Security for Windows XP Professional and Windows XP Home Edition](#)
- [Windows XP and .NET: An Overview](#)
- [PKI Enhancements in Windows XP Professional and Windows Server 2003](#)
- [Encrypting File System in Windows XP and Windows Server 2003](#)
- [Securing Mobile Computers with Windows XP Professional](#)
- [Authenticode for Internet Explorer 5.0](#)
- For the latest information about Windows XP, see the [Windows XP Web site](#).
- For the latest information about Windows Server 2003, see the [Windows Server 2003 Web site](#)

---

[Manage Your Profile](#)

© 2006 Microsoft Corporation. All rights reserved. [Terms of Use](#) | [Trademarks](#) | [Privacy Statement](#)

**Microsoft**