Symantec Endpoint Protection 11.0

Application and Device Control

Technical Product Management Team

Endpoint Security

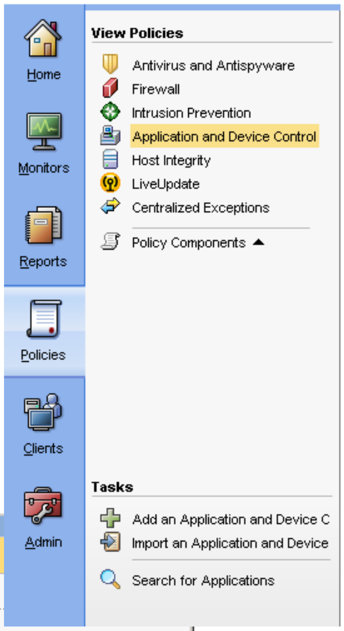
Revision 1.1

Application Control

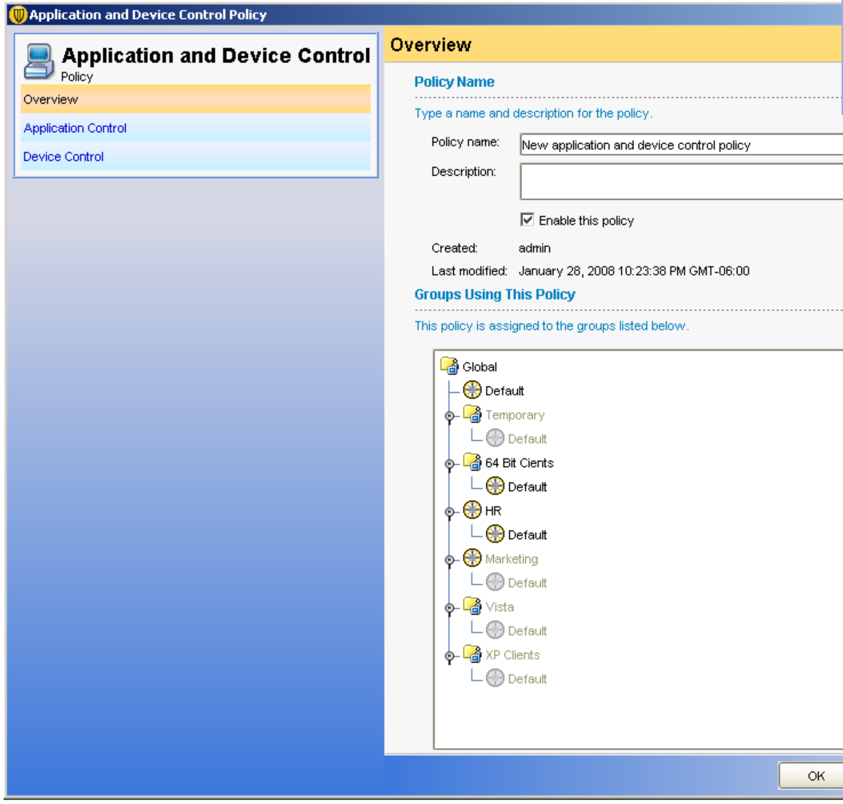
Application Control is an advanced security feature included in Symantec Endpoint Protection 11.0. Application Control provides administrators with the ability to monitor and/or control the behavior of applications. Administrators can grant/deny access to certain registry keys, files, and folders. In addition, administrators can also define which applications are permitted to run, which applications that cannot be terminated through irregular processes, and which applications can call Dynamic Link Libraries. Although Application Control provides administrators the ability to completely control the behavior of applications and users, many administrators have difficulty configuring Application Control Policies. This document will focus on providing several examples on how to create Application Control Policies to take full advantage of all the capabilities of this feature.

Creating an Application Control Policy

Application Control Policies can only be created and/or modified from the Symantec Endpoint Protection Admin console. Application Control cannot be modified on the Endpoint Protection Client. To manage Application Control policies:



1. Open the SEPM Console
2. On the Left Tab, Select Policies
3. Under the View Policies Tab, Select Application and Device Control
4. In the tasks section, select Add an Application and Device Control Policy


Application and Device Control Policy
Application and Device Control
Policy
Overview
Application Control
Device Control
Overview

Policy Name

Type a name and description for the policy.

Policy name: New application and device control policy

Description:

☑ Enable this policy

Created: admin

Last modified: January 28, 2008 10:23:38 PM GMT-06:00

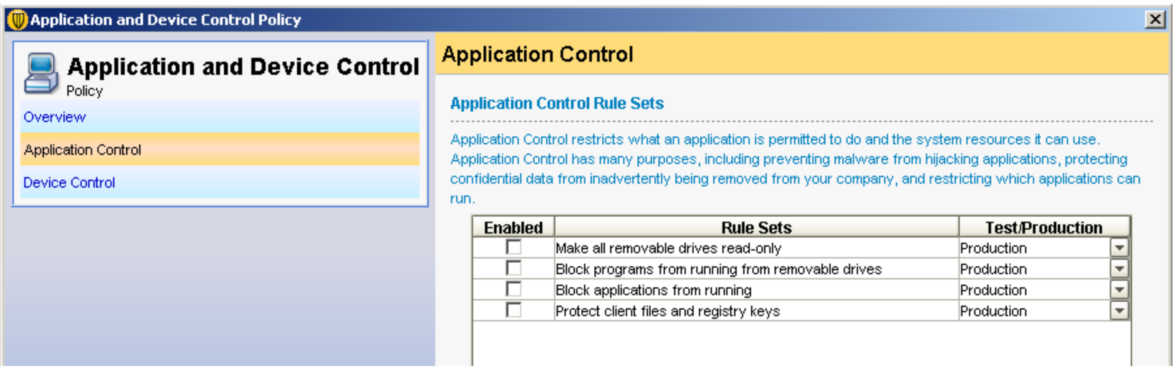
Groups Using This Policy

This policy is assigned to the groups listed below.

5. In the Policy Name Type the name you would like to give this policy
6. In the Description Field, enter a description for the policy
7. From the Left Menu, select Application Control

Application Control Rule Sets

A Rule Set is a set of controls that allow administrators to allow or block an action. In the example below, you will note that there are currently four rule sets defined. You will also notice that Administrators can choose to create as many rule sets as they would like in a policy. Even though multiple rule sets can be in a given policy, administrators can choose which rule sets are active by toggling the Enabled option. In this example, you will note that none of the rule sets are enabled.



Application and Device Control Policy
Application and Device Control
Policy
Overview
Application Control
Device Control

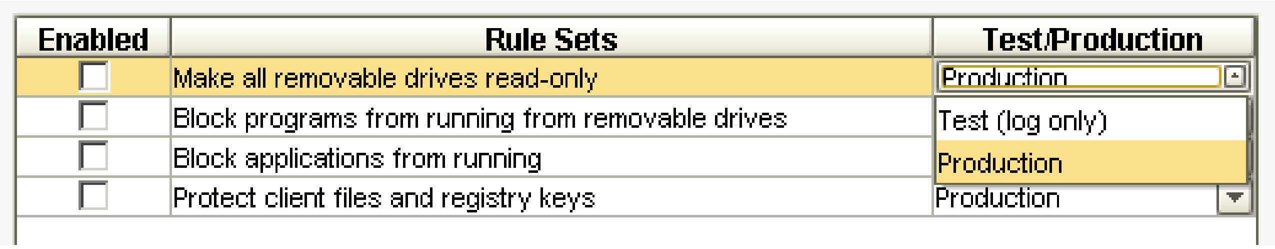
Application Control

Application Control Rule Sets

Application Control restricts what an application is permitted to do and the system resources it can use. Application Control has many purposes, including preventing malware from hijacking applications, protecting confidential data from inadvertently being removed from your company, and restricting which applications can run.

Enabled	Rule Sets	Test/Production
□	Make all removable drives read-only	Production
□	Block programs from running from removable drives	Production
□	Block applications from running	Production
□	Protect client files and registry keys	Production

To the right of the rule set name there is an option to configure Test/Production. This feature allows administrators to test rules before actually enabling them. In the Test (log only) configuration, no actions will be applied in the rule, but the action is logged. This allows administrators to see what would have happened if this rule would have been active. All new rule sets are created with the default option configured to test. This reduces potential accidents an administrator may make by not considering all possibilities of the rule.



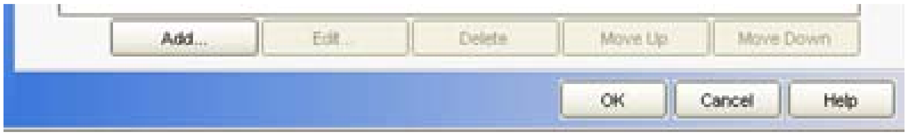
Enabled	Rule Sets	Test/Production
□	Make all removable drives read-only	Production
□	Block programs from running from removable drives	
□	Block applications from running	
□	Protect client files and registry keys	Production

BEST PRACTICE FOR RULE SETS:

It is recommended to run rules in a test mode for some acceptable period of time before switching them to production. During this time period it is recommended for you to review the logs and verify that the correct items are blocked.

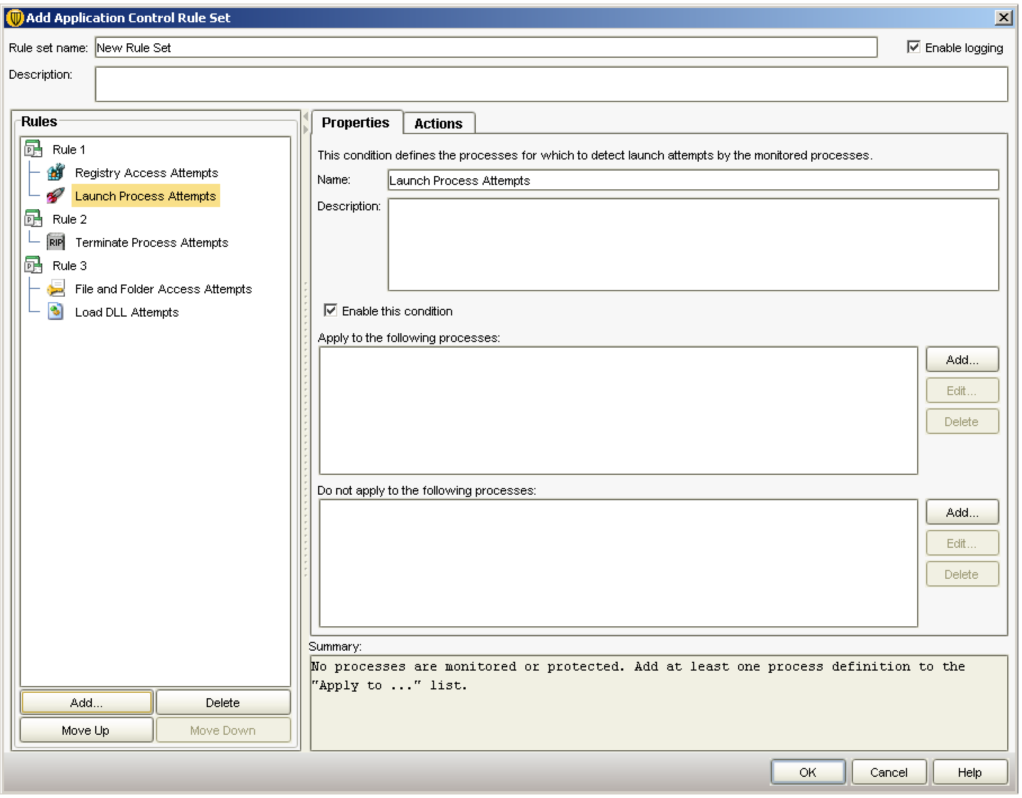
Rule Sets

To add a rule set, simply select the add button. To edit or delete a rule set, select the rule, then click the edit or delete button.



Note: The edit, delete, move up, and move down button will remain grayed out until a rule set is selected.

Rule Sets consists of Rules and Conditions. A rule is a set of conditions and actions that apply to a given process or processes. For organizational purposes, it is recommended to create a rule set that includes all of the actions that you want to allow/block/monitor a given task. For example, if an administrator wanted to block write attempts to all removable drives and block people/applications from tampering with a specific application, it would be recommended to create two distinct rule sets versus creating all of the necessary rules to do both tasks under one rule set.



Rule set name: New Rule Set ☑ Enable logging

Description:

This condition defines the processes for which to detect launch attempts by the monitored processes.

Name: Launch Process Attempts

Description:

☑ Enable this condition

Apply to the following processes:

Do not apply to the following processes:

Summary:

No processes are monitored or protected. Add at least one process definition to the "Apply to ..." list.

As you can see from the screen shots below, administrators can add as many rules and/or conditions as they want to a given rule set.

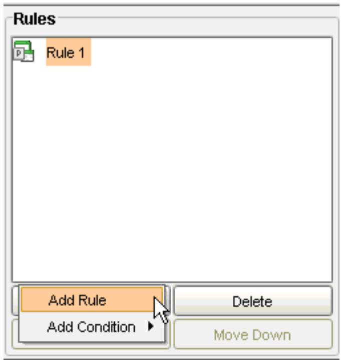
Rules

Rules define the application(s) that you are monitoring. Conditions define what specifically you want to allow or block an application from doing, and actions determine what action to take when the condition is meet.

Before getting into rule development, it is best to understand how rules work. A rule applies to an application or multiple applications. Rules contain conditions that monitor specified operations for the application(s) defined in the rule. The condition also contains the actions to take when the specified operation is observed. A majority of the issues encountered by new administrators when configuring Application Control is caused by not realizing that Actions always apply to the process defined in the rule and not the Condition.

BEST PRACTICE FOR RULES:

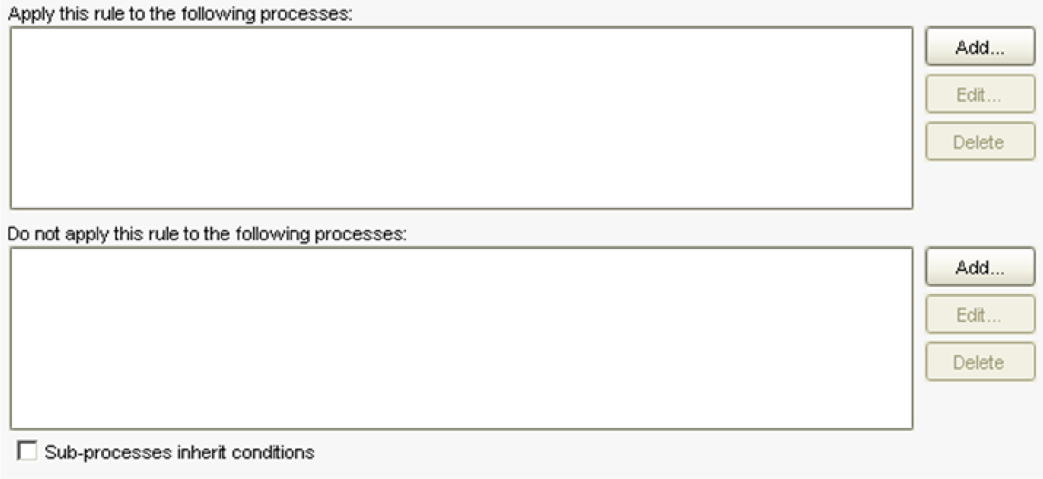
Actions always apply to the process defined in the rule, not by the Condition.



Adding a Rule

1. Open the Rule Set
2. Click the Add button in the Rules pane
3. Click the Add Rule option.
4. In the Rule Name, enter a name for the rule
5. In the Description Field, enter a description for the rule (optional)

Now that the rule has been created, the rule must be tied to an application or multiple applications. This is done in the Rules Properties window pane. You will also notice that you have an option to enable or disable the rule by toggling the Enable this Rule Check Box.


Apply this rule to the following processes:
Add...
Edit...
Delete
Do not apply this rule to the following processes:
Add...
Edit...
Delete

□ Sub-processes inherit conditions

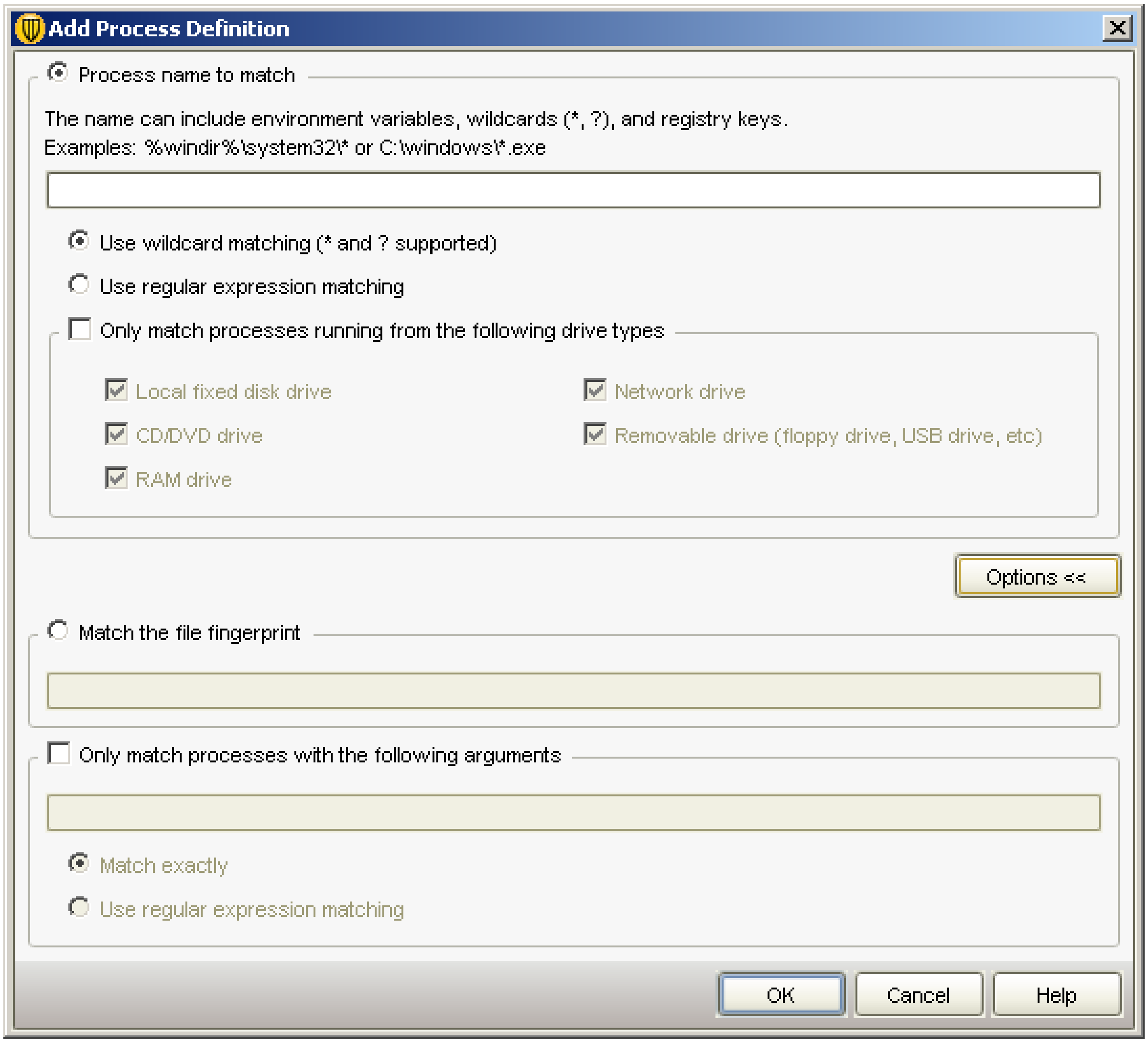
There are two sections that deal with tying the rule to an application or multiple applications. One process definitions list contains processes to which the rule applies. The other process definitions list contains processes to which the rule does not apply. If an administrator wanted to tie the rule to all application except for a given set of applications, then they would define a wildcard for all (*) in the top section, and list the applications that need an exception in the bottom section.

NOTE: In every configuration, the top section must have at least one application defined.

When adding applications to a rule, administrators can use the process name, wildcards, regular expressions, fingerprints, and/or drive types from where the application was launched.

Adding an Application to a Rule

1. Determine if the Application being added is the application to tie the rule to or if the application is going to be an exception to the rule
2. Select the Add for the appropriate the section



● Process name to match

The name can include environment variables, wildcards (*, ?), and registry keys.
Examples: %windir%\system32\* or C:\windows\*.exe

● Use wildcard matching (* and ? supported)

○ Use regular expression matching

□ Only match processes running from the following drive types

☑ Local fixed disk drive

☑ Network drive

☑ CD/DVD drive

☑ Removable drive (floppy drive, USB drive, etc)

☑ RAM drive

○ Match the file fingerprint

□ Only match processes with the following arguments

● Match exactly

○ Use regular expression matching

3. In the Add Process Definition, you can use the criteria of choice to define the application(s)

Administrators can define as many applications as they would like to a given rule.

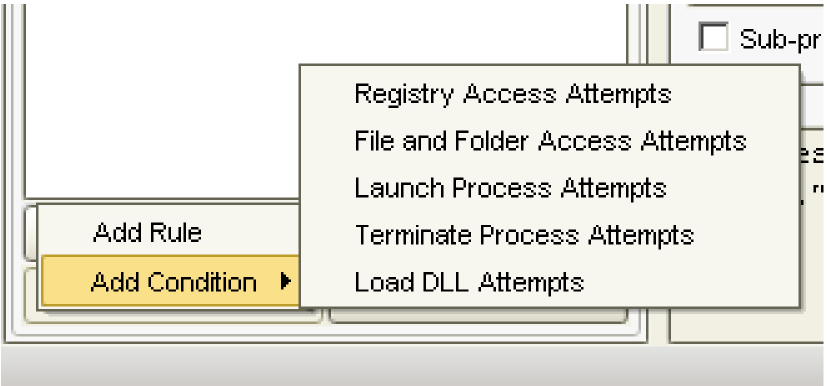
Conditions

Conditions are operations that can be allowed or denied for an application or multiple applications. There are several condition types that can be configured. These include the following:

- **Registry Access Attempts** - Allow or block access to a client computer's registry settings
- **File and Folder Access Attempts** - Allow or block access to defined files or folders on a client computer

- **Launch Process Attempts** -Allow or block the ability to launch a process on a client computer
- **Terminate Process Attempts** - Allow or block the ability to terminate a process on a client computer. For example, you may want to block a particular application from being stopped.
  NOTE: *This Condition does not prevent an application from being terminated using normal methods of quitting an application (i.e. Alt-F4, or the Program's native exit routine). It will prevent the process from being terminated by other applications or procedures.*
- **Load DLL Attempts** - Allow or block the ability to load a DLL on a client computer

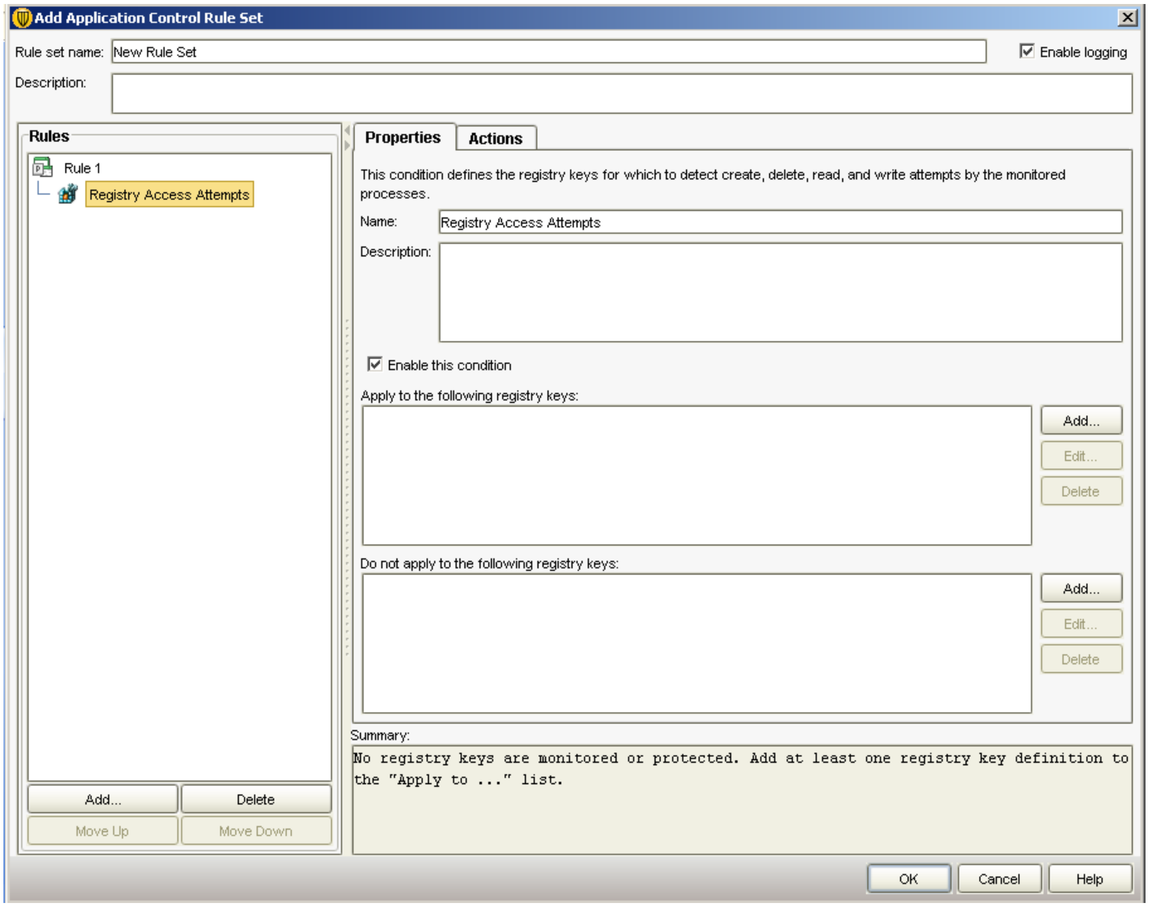
To add a condition to a rule, select the add Click in the Rules pane and then select the condition type you want to add.



Multiple conditions can be added to a given rule. As conditions are added, administrators will need to specify the specific properties of the condition and what actions to take when the condition is meet. Each condition type will have different properties.

Condition Properties

Each condition type has its own Condition Properties to specify what the condition is looking for. Each condition also has its own specific actions to take when the condition is true. To edit the properties and action for a condition, select the condition in the rules windows pane.



Add Application Control Rule Set

Rule set name: New Rule Set

☑ Enable logging

Description:

This condition defines the registry keys for which to detect create, delete, read, and write attempts by the monitored processes.

Name: Registry Access Attempts

Description:

☑ Enable this condition

Apply to the following registry keys:

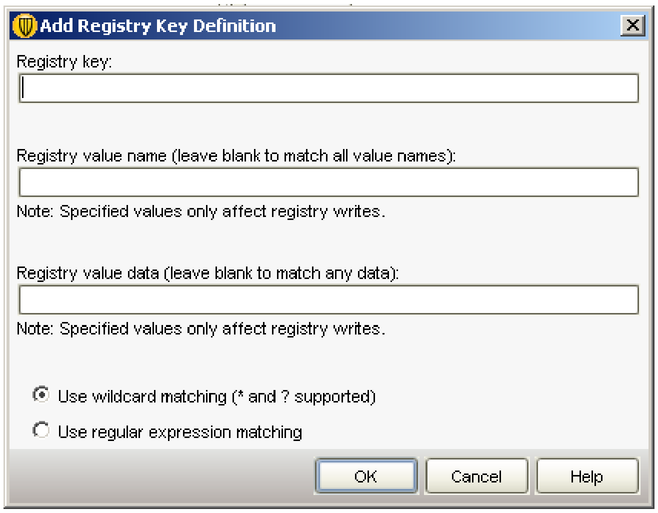
Do not apply to the following registry keys:

Summary:

No registry keys are monitored or protected. Add at least one registry key definition to the "Apply to ..." list.

Registry Access Attempts

Administrators can define specific registry keys, values, and data to monitor as depicted in the graphic below. Administrators can also use wildcards and regular expressions to define keys, values, and data.



Add Registry Key Definition

Registry key:

Registry value name (leave blank to match all value names):

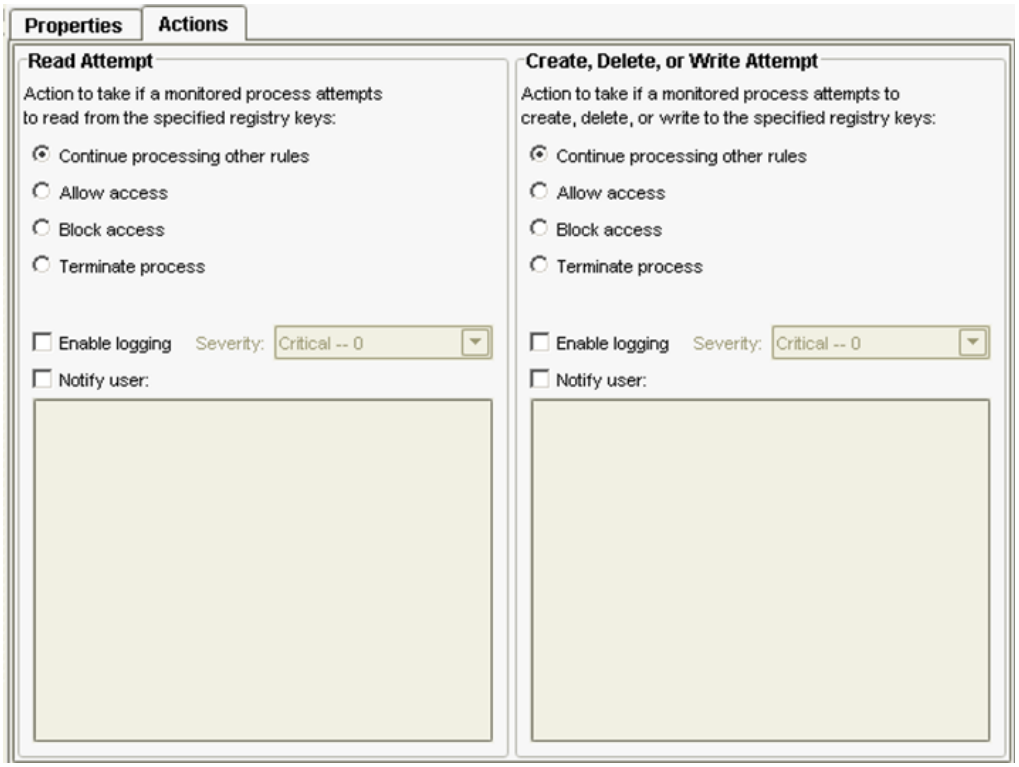
Note: Specified values only affect registry writes.

Registry value data (leave blank to match any data):

Note: Specified values only affect registry writes.

● Use wildcard matching (* and ? supported)

○ Use regular expression matching



Read Attempt

Action to take if a monitored process attempts to read from the specified registry keys:

● Continue processing other rules

○ Allow access

○ Block access

○ Terminate process

□ Enable logging Severity: Critical -- 0

□ Notify user:

Create, Delete, or Write Attempt

Action to take if a monitored process attempts to create, delete, or write to the specified registry keys:

● Continue processing other rules

○ Allow access

○ Block access

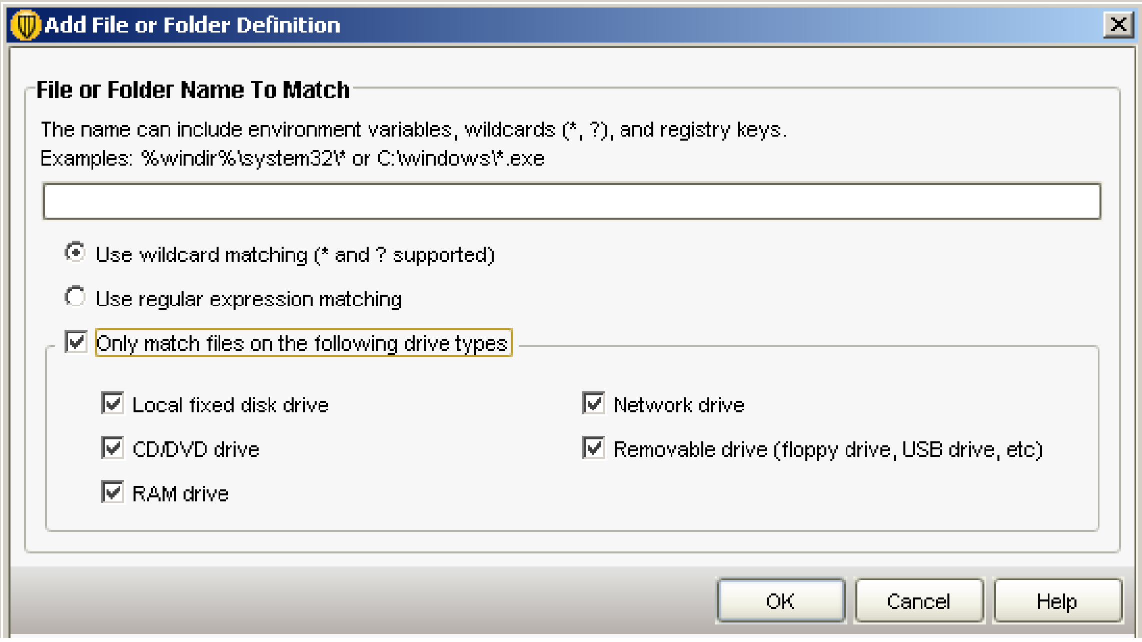
○ Terminate process

□ Enable logging Severity: Critical -- 0

□ Notify user:

For registry access attempts, administrators can define different actions to take for read and/or create/delete/write attempts.

File and Folder Access Attempts
Administrators can define files and folders to monitor as depicted in the graphic below. Administrators can also use wildcards and regular expressions. In addition, Administrators can also restrict the monitoring of files and folders to specific drive types.



Add File or Folder Definition

File or Folder Name To Match

The name can include environment variables, wildcards (*, ?), and registry keys.
Examples: %windir%\system32\* or C:\windows\*.exe

● Use wildcard matching (* and ? supported)

○ Use regular expression matching

☑ Only match files on the following drive types

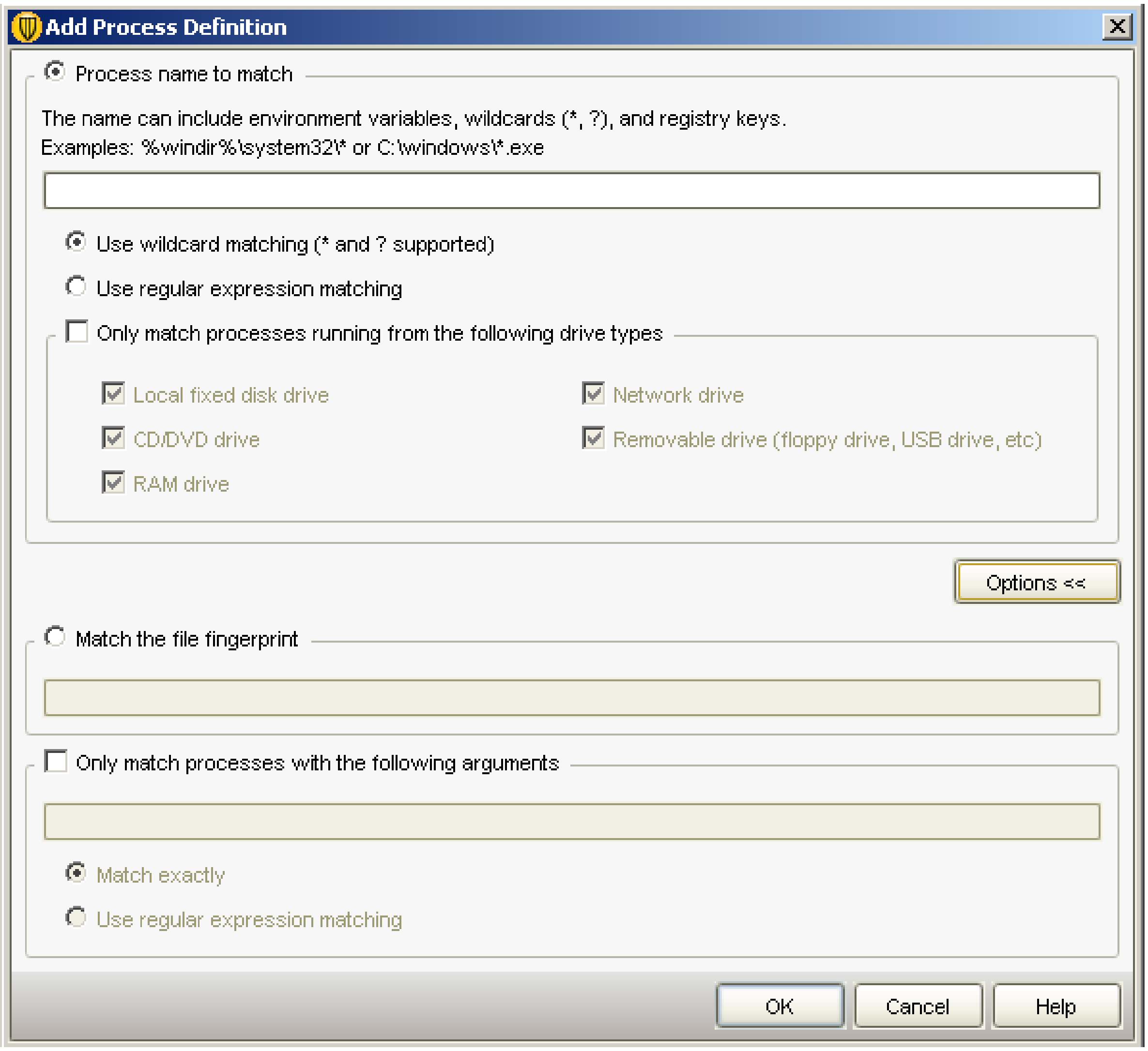
☑ Local fixed disk drive
☑ CD/DVD drive
☑ RAM drive
☑ Network drive
☑ Removable drive (floppy drive, USB drive, etc)

NOTE: When applying a condition to everything in a given folder, it is best to {folder name}*. In many cases, administrators forget to include the wildcard to include all files

Similar to Registry Attempt Access, administrators can choose to take different actions for read and/or Create/Delete/Write attempts as depicted below.

Launch Process Attempts

Administrators can define processes that they want to prevent or allow to start. When defining processes, administrators can use specific file names, wildcards, and regular expressions. Administrators can also choose to limit monitoring to applications being launched from a particular drive type, arguments being passed to an application, and or applications with a particular file fingerprint.



● Process name to match

The name can include environment variables, wildcards (*, ?), and registry keys.
Examples: %windir%\system32\* or C:\windows\*.exe

● Use wildcard matching (* and ? supported)

○ Use regular expression matching

□ Only match processes running from the following drive types

☑ Local fixed disk drive

☑ Network drive

☑ CD/DVD drive

☑ Removable drive (floppy drive, USB drive, etc)

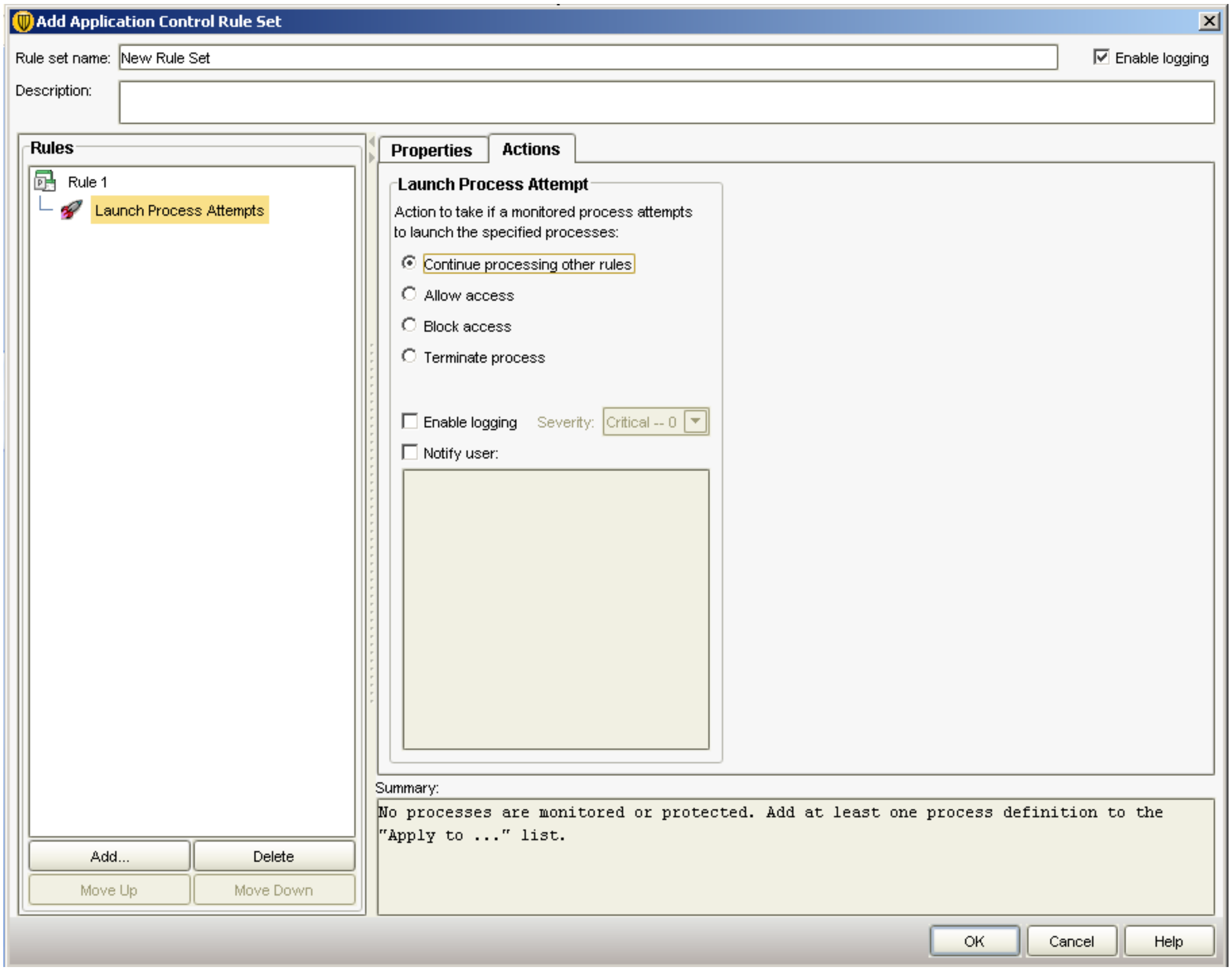
☑ RAM drive

○ Match the file fingerprint

□ Only match processes with the following arguments

● Match exactly

○ Use regular expression matching



Rule set name: New Rule Set ☑ Enable logging

Description:

Rule 1
Launch Process Attempts

Launch Process Attempt

Action to take if a monitored process attempts to launch the specified processes:

● Continue processing other rules
○ Allow access
○ Block access
○ Terminate process

□ Enable logging Severity: Critical -- 0
□ Notify user:

Summary:

```
No processes are monitored or protected. Add at least one process definition to the
"Apply to ..." list.
```

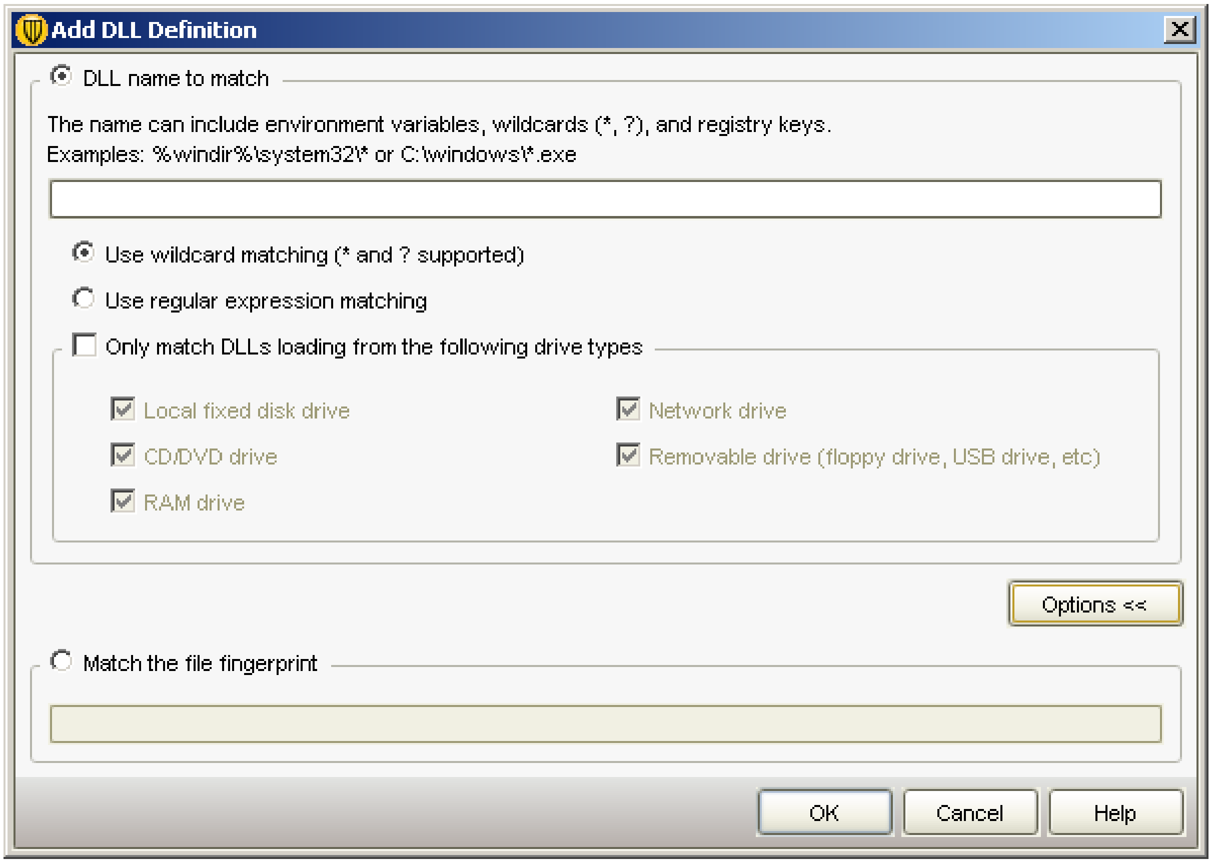
The actions for Launch Process Attempts are limited to allowing the process, blocking the process from being launched, or terminate the calling application.

Terminate Process Attempts

Properties and actions for Terminate Process Attempts are similar to Process Launch Attempts. The only difference between the two is that Terminate Process Attempts looks for applications/processes that try to kill a specified process vs. looking for a process to start.

Load DLL Attempts

Administrators can define Dynamic Link Library files that they want to prevent or allow to be loaded into an application. When defining DLLs, administrators can use specific file names, wildcards, fingerprint and regular expressions. Administrators can also choose to limit monitoring of DLLs to DLLs being launched from a particular drive type.



Add DLL Definition

● DLL name to match

The name can include environment variables, wildcards (*, ?), and registry keys.
Examples: %windir%\system32\* or C:\windows\*.exe

● Use wildcard matching (* and ? supported)

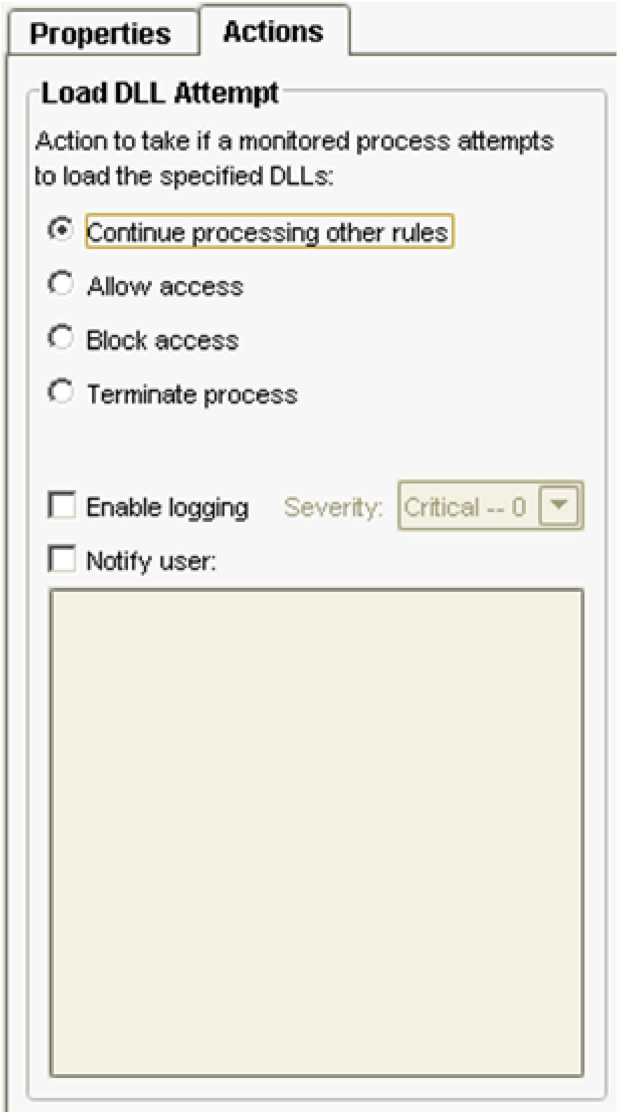
○ Use regular expression matching

□ Only match DLLs loading from the following drive types

☑ Local fixed disk drive
☑ Network drive
☑ CD/DVD drive
☑ Removable drive (floppy drive, USB drive, etc)
☑ RAM drive

○ Match the file fingerprint

In the Actions, administrators can choose to allow the DLL to load, block the DLL from being loaded, or terminate the application that is attempting to load the DLL.



Properties | Actions

Load DLL Attempt

Action to take if a monitored process attempts to load the specified DLLs:

● Continue processing other rules

○ Allow access

○ Block access

○ Terminate process

□ Enable logging Severity: Critical -- 0

□ Notify user:

Common Mistakes

There are two common mistakes made by individuals configuring Application Control for the first time. The first is configuring the wrong action and the second is neglecting the order of rules.

Wrong Action

In every action setting, there are four options for the action to take: Continue processing other rules, Allow access, Block access.

- **Continue processing** – This action allows administrators the ability to log the event and continue processing other rules in the stack. The standard operation is to stop processing rules once the first criteria matches.

- **Allow** – Allows the operation to continue
- **Block** – Prevents the operation
- **Terminate process** – Kills the application making the request.

Although these options seem simple, many people will accidentally choose to terminate the process. This can lead to undesired results. To fully understand the common mistake, consider the scenario below:

An Administrator wants to block individuals from modifying the secret.doc on client machines. The administrator does the following:

1. Creates a new Rule Set
2. Adds a new rule to the rule set
3. The rule is tied to the application *
4. Adds a condition, File and Folder Attempt Access
5. Adds the file secret.doc to the Condition
6. Configures the Write Action to Terminate Process

To test the policy, the administrator opens MS Word. The administrator then proceeds to use Word to navigate to the folder where secret.doc is located. The administrator opens the file. The Administrator makes some changes and then attempts to save the file. The End Result, MS Word terminates. Although no writes were allowed, the administrator did not expect MS Word to close. The reason this occurred is due to the Administrator choosing the Terminate option vs. the Block option.

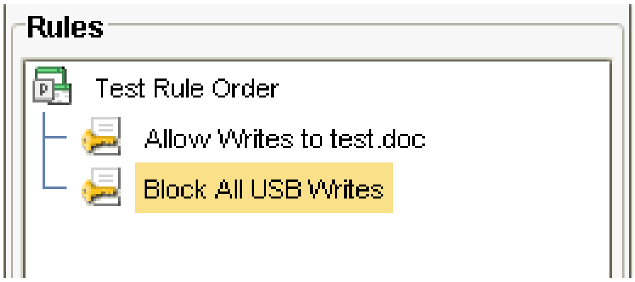
BEST PRACTICE FOR ACTIONS:

It is recommended to use the Block Action to prevent a condition vs. Terminate. Terminate should be only be used in advanced configurations.

Order of Rules

Many new administrators fail to notice that Application Control rules function very similar to the way that most network based firewalls work with the first rule match feature. What this means is that when there are multiple rules where the conditions are true, the rule list on the top will be the only condition/action that will be applied. This is unless they continue to process other rules action is set. It is important to understand the order of rules being configured. Neglecting the order could lead to wrong expectations. Consider the following scenario:

Suppose an administrator wanted to block everyone from moving/copying/creating files on USB drives, so the Administrator adds a rule to an existing rule set as depicted below.



Rules

Test Rule Order
- Allow Writes to test.doc
- Block All USB Writes

In the above scenario, clients would be able to create/modify a file called test.doc on USB drives. Because the Allow Writes to test.doc is ordered before the Block All USB Writes, the Block ALL USB Writes never gets processed in the case where rules above it are true.

Performance Impact

Note that when doing a block all for read & write actions, smcgui.exe can cause higher than normal CPU utilization. The workaround is to exclude smcgui.exe from this rule. This appears to impact systems limited RAM (512MB or less typically.)

Changes to Application and Device Control policies

Application and Device Control policies have been expanded to leverage the combination of information retrieved from the DevViewer and the tuned capabilities of Device Blocking.

To add a hardware device Application and Device Control policy

1 Click Add a policy in the Application and Device Control Policies dialog box.

2 Click the Device Blocking tab, and then click Add.

3 Continue with all of the other policy actions that you otherwise use.

■ Program definition, including processing an application rule, launching a process, or terminating a process

■ File definition, including regulating file access or loading a DLL

Note: Be aware that you can only block a device that is the end note of your hardware tree, unless the end node is "Generic volume" or "Storage volume." Thus if you have a specific USB drive that you want to block, you cannot block it by blocking all USB devices. You must be specific.

Common device types used in Application and Device Control

Most Device IDs that are supported with Application Control will have one of these types. In each case we show the generic type and a specific example of it.

DevViewer lets you query your system

■ USBSTOR

USBSTOR\DISK&VEN_SANDISK&PROD_CRUZER_MICRO&REV_2033\0002071406&0

■ FDC

FDC\GENERIC_FLOPPY_DRIVE\4&371082C9&0&0

■ IDE

IDE\DISKHTS541060G9SA00________________________MB3IC60H\4&14AA9DA8&0&0.0.0

■ SCSI

SCSI\DISK&VEN_WDC_WD50&PROD_00KS-00MNB0&REV_700.\4&1291CDED&0&000

Known issues

Application Control has some areas that are still under development.

- Application Control cannot block writing to CD/DVD drives.
- Application Control only blocking reading/writing devices that can be seen and accessed using Windows Explorer. If the device uses a third-party application to read or write, we cannot block that application. CD/DVD drives are an example of this.
- Application Control cannot block NetBIOS file shares. Thus a drive that is blocked on your local computer is still available if published as a NetBIOS share.

Adding Device Level Application and Device Control

Application and Device Control policies can also provide device level Application Control. Using device level Application Control, administrators can block or allow devices that attach to the system through interfaces, such as USB, inferred, firewire, SCSI, serial and parallel ports. Device level protection gives the administrator more control over what devices are allowed on their client computers.

For example, a mobile salesperson uses their personal USB key to store customer financial information. While traveling, the salesperson leaves the USB device at an airport lounge.

Unfortunately, the USB key was not password protected and exposed confidential customer information. To prevent information from leaking out of the organization, the company needs a solution that can prevent employees from using personal removable media devices to remove sensitive information from company computers and networks, while still allowing the transfer of information through secure removable media provided by the company (such as CD/DVD burners or PCMCIA hard drives).

In the following example, you will use device level Application Control to block all USB devices, yet allow mouse and keyboard devices. This would allow you to prevent the use of nonstandard devices such as joysticks, etc.

Normally, you would do this at the same time you were creating or updating the Application and Device Control Policy as described previously.

To add device level protection to the Application and Device Control policy in the Home location:

1. Make sure you’re still on the Policies tab.

2. In the Policies tree, select Global.

3. In the Home location, find the Harden Desktop Application and Device Control policy and click Edit on the right.

4. Click the Device Blocking tab. It consists of two parts: Blocked Devices and Devices Excluded from Blocking.

5. Under Blocked Devices, click Add, select USB devices, and click OK.

6. Under Devices Excluded From Blocking, click Add, select Human Input Devices (Mouse, Keyboard, etc.), and click OK.

7. Click Log blocked devices to add a log entry to the Agent’s Security log whenever a device is blocked.

8. When you’re done, click OK to close the Application and Device Control Policy Setting dialog box. The Application and Device Control policy is updated and downloaded to the Agent.

Testing Device Control

Remember that your Application and Device Control policy in the Home location is configured to block USB devices and to allow Human Input Devices. You can test it by attaching any USB device to the Agent computer. For example, you can try attaching a USB flash drive (or USB key).

To test device control:

1. Continuing to work on the Agent computer in the Home location, insert the USB flash drive into a free USB port.

2. Check to see whether the device is recognized. It is not available for use.

3. Double-click the Agent icon in the system tray to display the Agent window.

4. On the menu, choose Tools|Logs|Security Log. The Security Log appears and a Device Manager message appears. Select the message to display information in the bottom left window as shown here:

5. Reconnect the network cable, watch the location change back to Office, then you should see the USB device become available on the Agent computer.

You can now specify a new hardware device by Class ID or Device ID.
To add a new hardware device
1 In the Policy Library, click **Hardware Devices > Add a Device**.
The Add Device dialog box appears.

2 Type a device name and either a Class ID or a Device ID for the new device.

Class IDs
A Class ID is always a GUID. Examples:
■ Disk drives:
{4d36e967-e325-11ce-bfc1-08002be10318}
■ Storage volumes:
{71a27cdd-812a-11d0-bec7-08002be2092f}
■ USB devices:
{36FC9E60-C465-11CF-8056-444553540000}
■ DVD/CD-ROM:
{4D36E965-E325-11CE-BFC1-08002BE10318}
■ IDE:
{4d36e96a-e325-11ce-bfc1-08002be10318}
■ PCMCIA:
{4d36e977-e325-11ce-bfc1-08002be10318}

Device IDs

Device IDs are more readable, and appear in this format, usually:
<class>\<type>&<vendor>&<model>&<revision>\<serial number>
USBSTOR\DISK&VEN_SANDISK&PROD_CRUZER_MICRO&REV_2033\0002071406&0
For Device ID we support the wildcards "*" and "?":

- Asterisk (*) means zero or more of any character
- Question mark (?) means a single character of any value

Examples:

- Any USB storage device:

USBSTOR*

- Any USB disk:

USBSTOR\DISK*

- Any USB SanDisk drive:

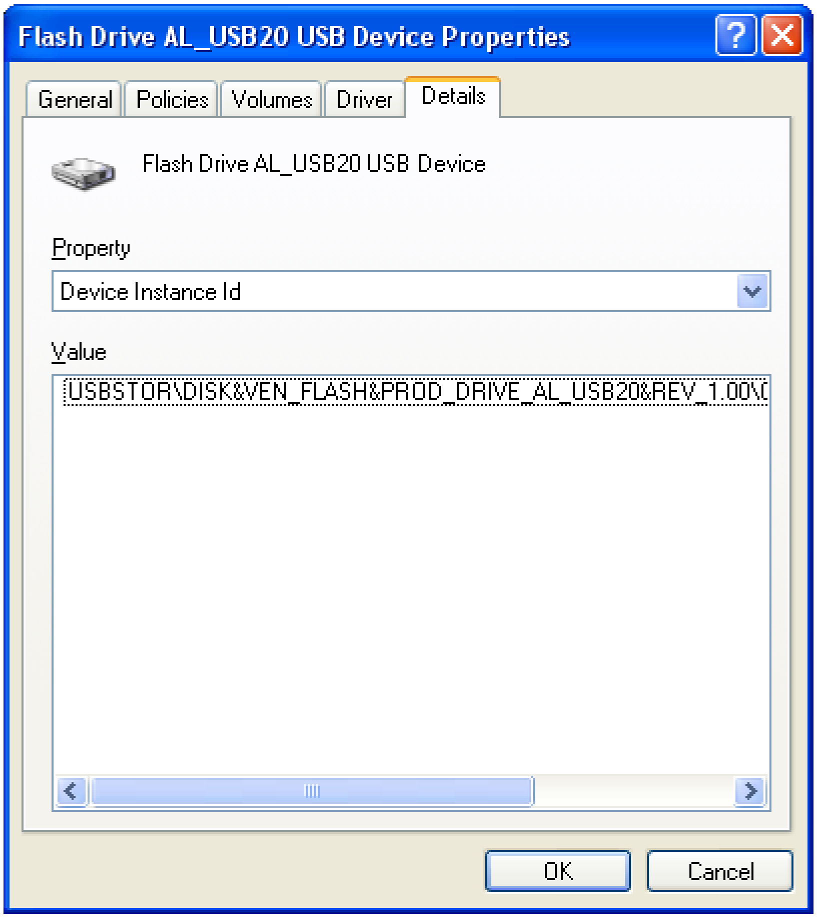
USBSTOR\DISK&VEN_SANDISK*

- Specific SanDisk device:

USBSTOR\DISK&VEN_SANDISK&PROD_CRUZER_MICRO&REV_2033\0002071406&0

- Specific Kingston device:

USBSTOR\DISK&VEN_KINGSTON&PROD_DTSECURE_PRIVACY*


Flash Drive AL_USB20 USB Device Properties
?
General
Policies
Volumes
Driver
Details
Flash Drive AL_USB20 USB Device
OK
Cancel

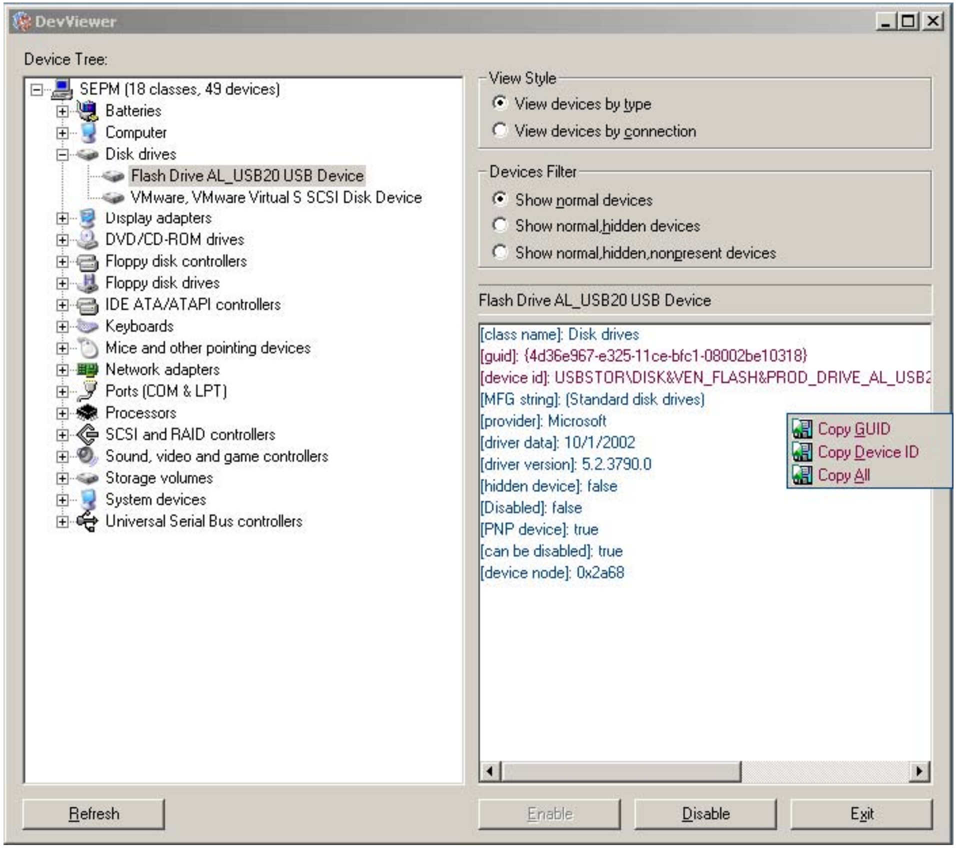
Property: Device Instance Id

Value: USBSTOR\DISK&VEN_FLASH&PROD_DRIVE_AL_USB20&REV_1.00\

Example of a Device ID found in the Details tab of the device properties

DevViewer lets you query your system

The DevViewer is an optional tool that lets you look at the devices on your system. You can list devices by type or by connection. You can then copy Class IDs and Device IDs to use in the creation of Application and Device Control policies. The user interface is very simple.



SEPM (18 classes, 49 devices)
- Batteries
- Computer
- Disk drives
  - Flash Drive AL_USB20 USB Device
  - VMware, VMware Virtual S SCSI Disk Device
- Display adapters
- DVD/CD-ROM drives
- Floppy disk controllers
- Floppy disk drives
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Network adapters
- Ports (COM & LPT)
- Processors
- SCSI and RAID controllers
- Sound, video and game controllers
- Storage volumes
- System devices
- Universal Serial Bus controllers

View Style
● View devices by type
○ View devices by connection

Devices Filter
● Show normal devices
○ Show normal,hidden devices
○ Show normal,hidden,nonpresent devices

Flash Drive AL_USB20 USB Device

```
[class name]: Disk drives
[guid]: {4d36e967-e325-11ce-bfc1-08002be10318}
[device id]: USBSTOR\DISK&VEN_FLASH&PROD_DRIVE_AL_USB2
[MFG string]: (Standard disk drives)
[provider]: Microsoft
[driver data]: 10/1/2002
[driver version]: 5.2.3790.0
[hidden device]: false
[Disabled]: false
[PNP device]: true
[can be disabled]: true
[device node]: 0x2a68
```

Device Blocking lets you block individual types of devices

The Device Blocking tab of the Application and Device Control Policy dialog box allows you to block devices by Device ID or by Class ID. You can also exclude devices from being blocked. In this way you can block a class of devices, but allow a subset of that class.