Using Expert Rules

Cb Protection v8.0.0
24 April 2017

Carbon Black, Inc.
1100 Winter Street, Waltham, MA 02451 USA
Tel: 617.393.7400 Fax: 617.393.7499
E-mail: support@carbonblack.com
Web: http://www.carbonblack.com

Table of Contents

Introduction

Custom, Memory and Registry Rules in Cb Protection provide built-in rule types to help you create rules suitable for many situations. Beginning with v8.0.0, Custom, Memory, and Registry Rules include a new “Expert” option. Expert rules, as the name implies, are intended for expert users. This new rule type exposes more operations you can use to trigger a rule and more actions you can take when the rule is triggered. It also allows you to combine multiple operations and actions into one rule.

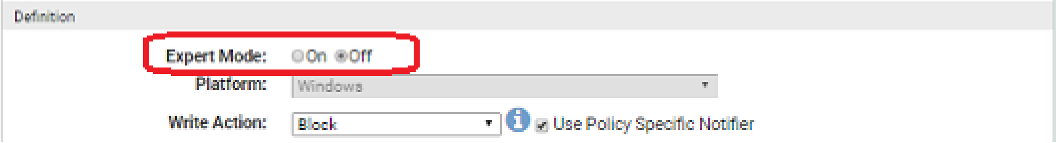
Important: Expert Rules do not have all of the error-checking that other rule types have, and it is possible to create unexpected (and unwanted) outcomes without being warned during rule creation. These rules are intended for use by Carbon Black Support or Services representatives, or customers working with them.

Enabling the Expert Rules Interface

Because Expert Rules offer a much longer list of options, the interface for creating them differs from that of standard rules. Instead of menus, Expert Rules are configured through checkboxes.

To open the Expert Rules interface for Memory or Registry Rules:

1. On the Memory or Registry Rules table page, click the **Add Memory Rule** or **Add Registry Rule** button.
2. On the Add Memory Rule or Add Registry Rule page, click the **On** radio button in the Expert Mode field in the Definition panel.



Definition

Expert Mode: ○On ●Off

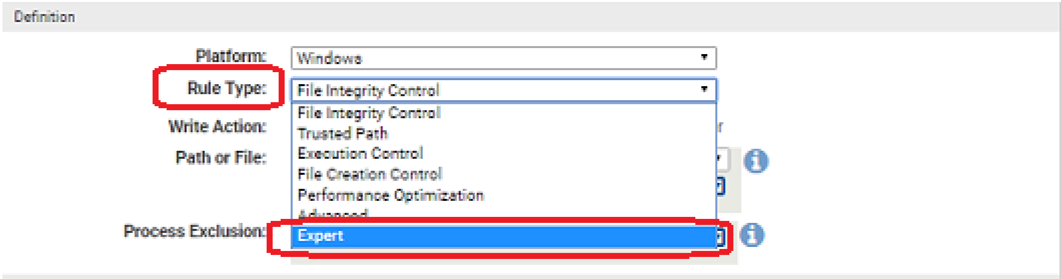
Platform: Windows

Write Action: Block

☑ Use Policy Specific Notifier

To open the Expert Rules interface for Custom Rules:

1. On the Custom Rules table page, click the **Add Custom Rule** button.
2. On the Add Custom Rule page, choose **Expert** on the Rule Type menu in the Definition panel.



Definition

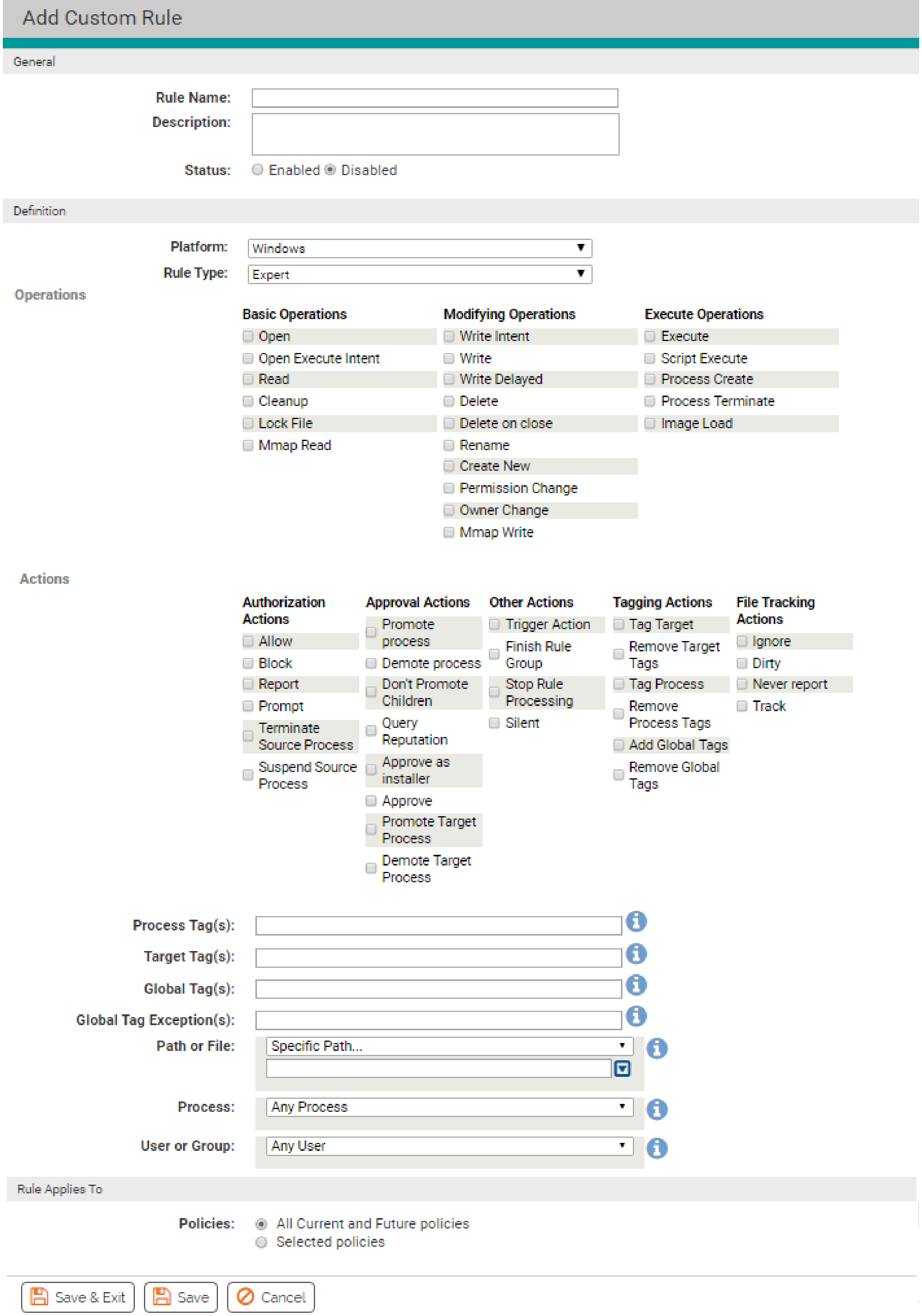
Platform: Windows

Rule Type: File Integrity Control

Write Action:

Path or File:

Process Exclusion:



Add Custom Rule

General

Rule Name:

Description:

Status: ○ Enabled ● Disabled

Definition

Platform: Windows

Rule Type: Expert

Operations

Basic Operations
- □ Open
- □ Open Execute Intent
- □ Read
- □ Cleanup
- □ Lock File
- □ Mmap Read

Modifying Operations
- □ Write Intent
- □ Write
- □ Write Delayed
- □ Delete
- □ Delete on close
- □ Rename
- □ Create New
- □ Permission Change
- □ Owner Change
- □ Mmap Write

Execute Operations
- □ Execute
- □ Script Execute
- □ Process Create
- □ Process Terminate
- □ Image Load

Actions

Authorization Actions
- □ Allow
- □ Block
- □ Report
- □ Prompt
- □ Terminate Source Process
- □ Suspend Source Process

Approval Actions
- □ Promote process
- □ Demote process
- □ Don't Promote Children
- □ Query Reputation
- □ Approve as installer
- □ Approve
- □ Promote Target Process
- □ Demote Target Process

Other Actions
- □ Trigger Action
- □ Finish Rule Group
- □ Stop Rule Processing
- □ Silent

Tagging Actions
- □ Tag Target
- □ Remove Target Tags
- □ Tag Process
- □ Remove Process Tags
- □ Add Global Tags
- □ Remove Global Tags

File Tracking Actions
- □ Ignore
- □ Dirty
- □ Never report
- □ Track

Process Tag(s):

Target Tag(s):

Global Tag(s):

Global Tag Exception(s):

Path or File: Specific Path...

Process: Any Process

User or Group: Any User

Rule Applies To

Policies: ● All Current and Future policies
○ Selected policies

Expert Rule Definitions

The basic components of an Expert Rule are the same as those of a non-expert rule: an **operation** that is being monitored, some combination of other conditions that must be met to match the rule, and an **action** to take when the rule is triggered. At least one operation and at least one action are required in an Expert Rule definition.

When multiple operations are defined in a rule, the rule triggers if *any* of them is true, as long as the action defined in the rule is possible for that operation.

Expert Rule Definition
If all other rule criteria are met **(source process, target file/path/process, user, policies)** …
… and if any of the ***Operations*** defined in the rule are attempted …
… then take the ***Actions*** defined in the rule (if available for the operation).

Expert Rule Operations

Expert versions of Custom, Memory, and Registry Rules each have their own set of Operations choices:

- Custom Rule Operations – see Table 1.
- Memory Rule Operations – see Table 2.
- Registry Rule Operations – see Table 3.

Table 1. Expert Custom Rules: Operation Settings

Column Name	Attempted Operation	Description
Execute Operations	Execute	Execution of a file
Execute Operations	Image Load	Loading of a file (dll, ocx, etc.) into memory
Execute Operations	Process Create	Creation of a new process
Execute Operations	Process Terminate	Termination of a process
Execute Operations	Script Execute	Execution of a script. For Cb Protection agents to see a script execute, the appropriate script rules should be defined in the console.
Modifying Operations	Delete On Close	This operation corresponds to someone opening/creating a file with the FILE_FLAG_DELETE_ON_CLOSE flag set (meaning someone intends to delete this file). Typically used for short-lived files, but also can be used by malware as an alternative way of deleting files.
Modifying Operations	Create New	Creation of a new file or directory
Modifying Operations	Delete	Deletion of a file
Modifying Operations	Mmap Write	Write to a memory mapped file

Column Name	Attempted Operation	Description
Modifying Operations	Owner Change	Change the owner of a file or directory
Modifying Operations	Permission Change	Change the permissions on a file or directory
Modifying Operations	Rename	Rename a file or directory
Modifying Operations	Write	Write the contents of a file; unlike other rules, "write" in the Expert Rules interface does not mean any modification at all. However, operations such as modifying the length of a file are also considered writes. **Note:** If you specify a rule that allows creation of new files but blocks writes to existing files, the agent will allow the process that created a new file to make modifications to that same file for a short period of time. Without this, the process that created the file could not write the initial content, and you would be left with a zero-byte file.
Modifying Operations	Write Delayed	Memory mapped writes in which an application "maps" a file into memory (RAM) and writes to that memory. This content is later flushed to disk by the operating system's paging mechanism.
Modifying Operations	Write Intent	Get a handle to a file with the intent to execute, but writing has not happened yet.
Basic Operations	Cleanup	Cleanup is the file system reporting that a process is done using a file; in essence, it means the file has been "closed" (Corresponds to IRP_MJ_CLEANUP; see https://msdn.microsoft.com/en-us/library/windows/hardware/ff548608(v=vs.85).aspx). In Cb Protection, cleanup signals that a file is ready to be analyzed, and is also the triggers file deletion in a "delete on close" operation. **Note:** Cleanup cannot be blocked since doing so would result in a handle leak. You can choose a reporting action, however.
Basic Operations	Lock File	Lock the file that matches the rule.
Basic Operations	Mmap Read	Read a memory mapped file.
Basic Operations	Open	File open action.
Basic Operations	Open Execute Intent	A file handle was acquired with the intent to execute, but execution has not happened yet.
Basic Operations	Read	Read the contents of a file.

Table 2. Expert Memory Rules: Operation Settings

Column Name	Attempted Operation	Description
Basic Operations	Access Kernel Memory	Rules can use this operation to close a bypass on XP and 2003 systems that prevents usermode processes from opening \Device\PhysicalMemory, which effectively allows them to read kernel memory. Windows versions from Vista forward prevent this action on their own.
Basic Operations	Allocate Memory	Corresponds to the VirtualAlloc system call, which is invoked when an application wants to obtain a block of memory with specific permissions.
Basic Operations	Debug Process	Corresponds to OB_OPERATION_PROCESS_PTRACE which is invoked when someone attempts to enable ptrace logging on another application.
Basic Operations	Kill Process	Corresponds to OB_OPERATION_PROCESS_KILL, which is used to signify that someone opened a handle to another process/thread and attempted to terminate it.
Process/Thread Operations	Create Handle	This operation occurs when someone is trying to open a new handle to a process/thread. Rules can be used to strip or report the permissions on the handle to limit what the source process can do on the target object. The permissions available here are the same as those documented for non-expert memory rules in the "Memory Rules" chapter of *Using Cb Protection.*
Process/Thread Operations	Duplicate Handle	This operation occurs when someone is trying to duplicate a handle that is already open. Rules can strip or report the permissions on the new handle to limit what the process can do with the duplicate handle.

Table 3. Expert Registry Rules: Operation Settings

Column Name	Attempted Operation	Description
Key Operations	Create Key	Creation of a registry key
Key Operations	Delete Key	Deletion of a registry key
Key Operations	Open Key	Open a registry key
Key Operations	Rename Key	Not implemented. Do not use. Rename is a delete operation plus a create operation. If you want to block or report renaming of keys, you can take the action on either of those operations.
Key Operations	Set Security	SetSecurity is invoked when someone tries to change the permissions on a given registry key/value.
Value Operations	Change Value	Change value is invoked when there is an attempt to modify a registry value. The target name of the operation is the full path to the registry value (e.g. HKLM\key\value).

Column Name	Attempted Operation	Description
Value Operations	Delete Value	Delete value is invoked when there is an attempt to delete a registry value. The target name of the operation is the full path to the registry value (e.g. HKLM\key\value).

Expert Rule Actions

Table 4 shows the actions an Expert Rule can take if an operation matching the rule is attempted. Unlike operations, most (but not all) actions are common to all three rule pages. The *Where Available* column in the table shows whether the action is limited to one page.

As with non-expert rules, Expert Rules are often most effective in pairs. For example, one rule might tag certain types of files and another one might take a specified action, such as allowing execution, when files with that tag appear later. Tags and Tagging Actions in Expert Rules on page 11 provides more information about this feature.

With Expert Rules, you can also combine actions that might otherwise require two rules. For example, you can configure rule to “promote” a process so that files it writes are locally approved, and in the same rule, demote children of the process so that files *they* write are not locally approved. When you review the table, look for actions that form this kind of pairing.

The table includes brief descriptions of what these actions do. Many of the actions are described in more detail in the “Custom Rules,” “Memory Rules,” and “Registry Rules” chapters in *Using Cb Protection,* which is available as online help in the v8.0.0 Cb Protection console or as a PDF download on the Carbon Black User eXchange.

Note: In the current release of v8.0.0, the Actions column does not show any values.

Mutually Exclusive Actions

Some of Expert Rule actions are mutually exclusive. If you choose one of the options in the following list, the others will be greyed out on the page:

- Allow, Block, Report, Prompt
- Promote process, Demote process
- Tag Target, Remove Target Tags, Tag Process, Remove Process Tags, Add Global Tags, Remove Global Tags
- Ignore, Dirty, Never Report, Track

Table 4. Action Settings in Expert Rules

Column Name	Setting Name	Description	Where Available
Approval Actions	Approve	Locally approve the target (file). **Note:** Currently, you cannot disable sending approval events in an Expert Rule. If you do not want an Expert Rule to send approval events, first create it as a non-expert rule, turn off *Send Approval Event*, save the rule, and then change it to an Expert Rule to finish your configuration.	Custom Rules
Approval Actions	Approve As installer	Locally approve the target (file) and mark it as an installer.	Custom Rules
Approval Actions	Demote process	Demote the process that performed the operation.	All
Approval Actions	Demote Target Process	Demote the target process.	Custom Rules
Approval Actions	Don't Promote Children	Do not promote child processes of the process that performed the operation; used when the process itself is promoted (see below).	All
Approval Actions	Promote process	Promote the process that performed the operation, locally approving *files* written by this process; promote new *processes* spawned by this process unless "Don't Promote children" was also chosen (see above).	All
Approval Actions	Promote Target Process	Promote the target process when this operation happens. Only applicable with the "Create process" operation.	Custom Rules
Approval Actions	Query Reputation	Ask server for the global state of the target (file) when this operation happens. This setting is for built-in rules and should not be activated in new rules or changed in existing rules.	All
Authorization Actions	Allow	Allow the corresponding operations to go through. **Note:** You can create an Expert Rule that allows creation of new files but blocks writes to existing files. However, the agent will allow the process that created a new file to make further modifications to that same file for a short period of time. This is necessary to allow the same process to both create the new file and write the initial content to the file.	All
Authorization Actions	Block	Block the corresponding operation.	All
Authorization Actions	Prompt	Prompt the user to decide whether to allow or block the operation. A notifier must be selected when this action is chosen.	All
Authorization Actions	Report	Report (as an event) that the operation would have been blocked, but do not block it.	All

Column Name	Setting Name	Description	Where Available
		Example: Generate an event for all new files written by Powershell.	
Authorization Actions	Suspend Source Process	Suspend the process that performed this operation. This is typically used for malware research where a researcher might want to inspect the process and see what it did or what it was about to do before it is terminated.	All
Authorization Actions	Terminate Source Process	Terminate the process that performed this operation	All
Tagging Actions	Tag Process	Tag the process; if chosen, one or more tags must be provided in the "Tags to Add/Remove" field	All
Tagging Actions	Tag Target	Tag the target object; if chosen, one or more tags must be provided in the "Tags to Add/Remove" field	All
Tagging Actions	Remove Process Tags	Remove tags from the process; if chosen, one or more tags must be provided in the "Tags to Add/Remove" field	All
Tagging Actions	Remove Target Tags	Remove tags from the target object; if chosen, one or more tags must be provided in the "Tags to Add/Remove" field	All
Tagging Actions	Remove Global Tags	Remove global tags that other rules can test; if chosen, one or more tags must be provided in the "Tags to Add/Remove" field	All
Tagging Actions	Add Global Tags	Add global tags that other rules can test; if chosen, one or more tags must be provided in the "Tags to Add/Remove" field	All
File Tracking Actions	Dirty	Trigger re-analysis of the file matching the Path or File definition to see whether its hash has changed	Custom Rules
File Tracking Actions	Ignore	Do not track modifications	Custom Rules
File Tracking Actions	Never report	Keep an agent record of these operations but do not them to the server	Custom Rules
File Tracking Actions	Track	Track the file regardless of ignore rules	Custom Rules
Other Actions	Finish Rule Group	Skip other user-created rules but continue evaluating all built-in rules.	All
Other Actions	Report Execution (Deprecated)	Trigger meter on first execution events; this field is deprecated and appears only when details of an internal rule are displayed. It is read-only.	Custom Rules
Other Actions	Silent	Perform all assigned rule actions, but don't generate notifiers or report events.	All
Other Actions	Stop Rule Processing	Stop processing other rules after this rule is processed, which may improve performance.	All

Column Name	Setting Name	Description	Where Available
		Note that Allow also stops processing but allows the action to continue.	
Other Actions	Trigger Action	Trigger agent action where usermode sends an event in response to a kernel operation. For use with internal rules only.	All
Other Actions	Unenforceable	Indicate that some other action could not be enforced due to platform limitations. This field appears only when details of an internal rule are displayed. It is read-only.	All

Tags and Tagging Actions in Expert Rules

Tags are labels that can be applied to different objects tracked in Cb Protection for as long as those objects exist. An “object” in this case can be a running process, a registry key, a file, an image, or the entire global ‘system’ that those processes run on. The global system is the computer the process is running on.

Each operation has an "initiator" process, the process that initiated it. Each operation also has a target object that the operation is being carried out on. Target objects vary depending on the type of operation. For example:

- For the "file write" operation, the target object will be a file.
- For a "process start” operation, the target will be another process.
- For a "registry value creation" operation, the target is a registry value. The behavior and lifespan of the tag depends on the *type* of object being tagged.

On the Add or Edit Rule page, the Tagging Actions column provides options for adding and removing tags when the other conditions of the rule are met. There are separate ‘add’ and ‘remove’ options for initiator processes, target processes, and the global system.

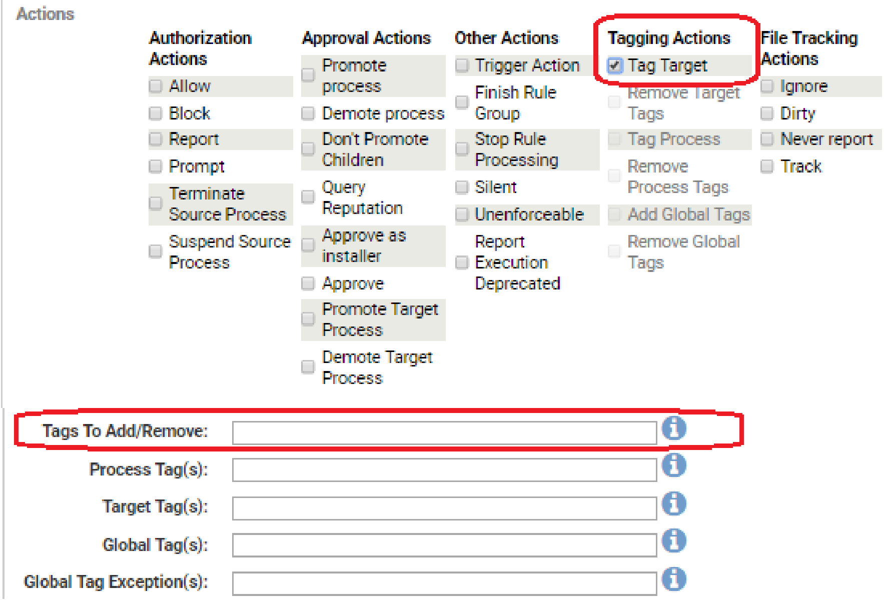
Tags are primarily useful when several rules related to the same tag(s) are created. Once a rule applies tags to an object, other rules can use these tags as a factor in determining whether a process matches the rule conditions, taking an action when a match is found. In other words, to use tags:

- Create a rule that applies a tag to an object.
- Create a separate rule that uses the presence of that tag as a condition for matching the rule; if it is testing the same operation as the tagging rule, rank this rule lower.

To use an Expert Rule to apply tags to an object:

1. On the Add Custom/Memory/Registry Rule page, provide a name for the rule.
2. If the rule name does not include the tag(s) you intend to use, consider putting them into the Description field. Although you cannot add a “tags” column on the rules table pages, you can display the description.
3. Choose Expert as the Rule Type (Custom Rules) or click the Expert Mode radio button (Memory and Registry Rules).

4. In the Operations list, choose the operation(s) that should trigger this rule.
5. In the Tagging Actions list, choose the object that you want to tag (Tag Target, Tag Process, Add Global Tags). When you choose one of these actions, the *Tags to Add/Remove* field is added below the list.



Actions

Authorization Actions
☐ Allow
☐ Block
☐ Report
☐ Prompt
☐ Terminate Source Process
☐ Suspend Source Process

Approval Actions
☐ Promote process
☐ Demote process
☐ Don't Promote Children
☐ Query Reputation
☐ Approve as installer
☐ Approve
☐ Promote Target Process
☐ Demote Target Process

Other Actions
☐ Trigger Action
☐ Finish Rule Group
☐ Stop Rule Processing
☐ Silent
☐ Unenforceable
☐ Report Execution Deprecated

Tagging Actions
☑ Tag Target
☐ Remove Target Tags
☐ Tag Process
☐ Remove Process Tags
☐ Add Global Tags
☐ Remove Global Tags

File Tracking Actions
☐ Ignore
☐ Dirty
☐ Never report
☐ Track

Tags To Add/Remove:

Process Tag(s):

Target Tag(s):

Global Tag(s):

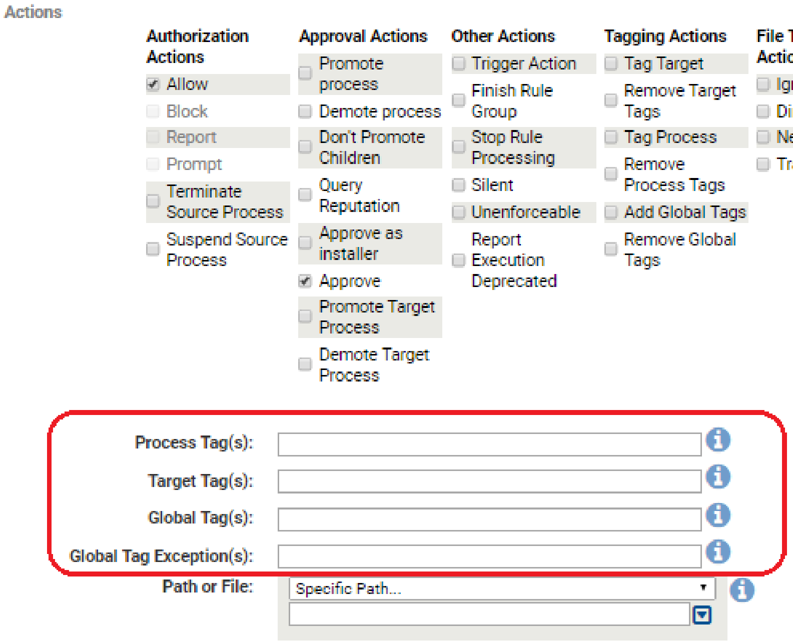
Global Tag Exception(s):

6. In the Tags to Add/Remove field, enter the name(s) of the tag(s) you want to apply when the conditions of this rule are met. To add more than one tag, separate the tag names with commas.
7. Provide any additional conditions for matching this rule, such as paths or files, the processes and any restrictions by user or policy.
8. When you have finished specifying the rule, click the **Save & Exit** button.

To create an Expert Rule that is applied to operations that have a specific tag:

1. Navigate to the table page for the kind of rule you want to create (Custom, Memory or Registry).
2. Click the add rule button.
3. On the Add Rule page, provide a name for the rule.
4. If the rule name does not include the tag(s) you intend to use, consider putting them into the Description field. Although you cannot add a "tags" column on the rules table pages, you can display the description. This will be helpful in pairing the rule that creates a particular tag with a rule that uses that tag to identify matching operations.
5. Choose Expert as the Rule Type (Custom Rules) or click the Expert Mode radio button (Memory and Registry Rules).
6. In the Operations list, choose the operation(s) that should trigger this rule.
7. In the Actions list, choose the action you want to perform when an operation matches the rule.
   Note: You do not need to use one of the Tagging Actions in this case unless you are using one tag to create another one.

8. Enter the names of the tags you want to match in the appropriate field(s):
   - **Process Tag(s):** Enter tags here if you want to apply this rule when the process that *initiates an operation* has a matching tag.
   - **Target Tag(s):** Enter tags here if you want to apply this rule when the process, file, or registry key that is the *target of an operation* has a matching tag.
   - **Global Tag(s):** Enter tags here if you want to apply this rule when the *'global system' on which the operation is being performed* has a matching tag. This is equivalent to the computer on which the operation is performed.
   - **Global Tag Exceptions(s):** Enter tags here if you want to *exclude* 'global systems' with any of the matching tags from being subject to this rule.



Actions

Authorization Actions
- ☑ Allow
- □ Block
- □ Report
- □ Prompt
- □ Terminate Source Process
- □ Suspend Source Process

Approval Actions
- □ Promote process
- □ Demote process
- □ Don't Promote Children
- □ Query Reputation
- □ Approve as installer
- ☑ Approve
- □ Promote Target Process
- □ Demote Target Process

Other Actions
- □ Trigger Action
- □ Finish Rule Group
- □ Stop Rule Processing
- □ Silent
- □ Unenforceable
- □ Report Execution Deprecated

Tagging Actions
- □ Tag Target
- □ Remove Target Tags
- □ Tag Process
- □ Remove Process Tags
- □ Add Global Tags
- □ Remove Global Tags

File Tracking Actions
- □ Ignore
- □ Dirty
- □ Never report
- □ Track

Process Tag(s):

Target Tag(s):

Global Tag(s):

Global Tag Exception(s):

Path or File: Specific Path...

9. Provide any additional conditions for matching this rule, such as paths, file names, processes, and any restrictions by user or policy.
10. When you have finished specifying the rule, click the **Save & Exit** button.

Tag Syntax Requirements

Tags must meet the following requirements and restrictions:

- Commas are used to separate tags when multiple tags are used; do not use commas in the tag name itself.
- Tags must have at least one non-numeric character.
- The prefixes "<Bit9:", "YaraTags" or "<LegacyClassification:" are reserved for use by Carbon Black and should not be used in a tag unless advised by Carbon Black Support or Services. See Built-in Tags on page 14 for more information.
- A tag and the process pattern of a rule (i.e., the pattern in any of the process fields) should not be the same. This helps avoid conflicts during rule processing.
- Avoid extremely long tag names. All of the fields in a rule combined must not exceed 2048 characters.

- Do not use the pipe (|) character.

Built-in Tags

Certain special tags are used by Cb Protection in some Custom Rules provided with this release. You should not use these tags unless advised otherwise by Carbon Black support or services.

- **<YaraTags:*tagname*>** – Yara tags come from Yara rule content. When used to match the process that initiated an operation or the target process, they refer to the *file* that the process's image was loaded from. Yara tag values persist as part of the tracked file state, including across reboots (unlike user-created tags).
  Direct customization of YARA rules is not supported in this release, and these tags are currently for use by Carbon Black only.
- **Bit9:***tagname* – The 'Bit9:' prefix is used on tags built-in to Cb Protection for various purposes. Although it is possible to use them in other rules, they are intended only for rules provided by Carbon Black, and their behavior could changed in later releases without notice.
- **<LegacyClassification**:*tagname*> – This prefix is used for internal, Carbon Black rules to identify older, hexadecimal tags. It should not be used in other rules.

Tag Persistence

User-created tags, for processes and for the global system, do not persist across reboots of an agent. The rule that attaches the tag must detect the operation it describes and reattach the tag before a rule that uses the tag can discover the tagged process. A tag may also be explicitly removed by a rule that has a "remove tag" action defined. There are other conditions that affect tags on different objects:

- **Process/thread tag:** Process and thread tags persist until the process instance dies. If the agent process (parity.exe) is restarted, then the tags would still be set. If the full system is restarted or if the kernel filter driver (parity.sys) is unloaded and reloaded, then a process would lose its classifications.
- **File tag:** Currently, a file tag lives only during a single operation.
- **Yara Tag:** Yara tags persist for the life the hash they apply to in the agent cache.
- **Global Tag:** Global tags persist until the agent process (parity.exe) is restarted.

Expert Rule Examples

Several of the default Custom Rules included in v8.0.0 are Expert Rules. You can examine the following rules to get ideas about the kind of rules you might choose to create:

- Examine powershell script contents
- Block powershell scripts that execute memory
- Do not treat these processes as .NET applications
- Report read-only memory map operations on unapproved executables by .NET applications
- [Sample] Prompt for read-only memory map operations on unapproved executables by .NET applications in medium enforcement
- [Sample] Deny read-only memory map operations on unapproved executables by .NET applications in high enforcement
- Deny read-only memory map operations on banned executables by .NET applications

Note: Registry or Memory Rules included by default in this release do not use Expert Mode.

Example: Allow Execution in a Folder when Visual Studio is Running

Perhaps you want to restrict executions in a specific folder, called *projectfolder* in the example here, so that they are allowed only when Visual Studio is running. This can be done using a series of Custom Rules to create, use, and remove that tag. Global tags essentially tag the entire environment on a computer indicating that anything happening on it matches the tag.

If you create a suite of rules like this, be sure to name them in a way that makes clear their relationship, and consider providing more information about their interactions in the Description field for each one. You can also further refine the rules with the other standard options, such as specifying user and/or policy.

1. Create one Custom Rule that applies a global tag when it detects Visual Studio running. For example:
   - **Operations:** Process Create
   - **Actions:** Add Global Tags
   - **Tags to Add/Remove:** VSwrite2projectfolder
   - **Target:** devenv.exe
2. Create a second Custom Rule that allows execution in a specific folder when the global tag is set. For example:
   - **Operations**: Execute
   - **Actions:** Allow
   - **Global Tag(s):** VSwrite2projectfolder
   - **Path or File:** <ProgramData>\projectfolder\
3. Create a third Custom Rule that removes the global tag when the Visual Studio process terminates. For example:
   - **Operations:** Process Terminate
   - **Actions:** Remove Global Tags
   - **Tags to Add/Remove:** VSwrite2projectfolder
   - **Target:** devenv.exe

Example: Tag a Process and Report its Children

Perhaps you want to tag all processes that are launched by svchost.exe so that you can report when the child process are running. You can create a pair of rules for this purpose.

As in the previous example, name the rules in a way that makes clear their relationship, and consider providing more information in the Description field for each one.

1. Create one Custom Rule that applies a tag to a process if it is the launched by svchost.exe. For example:
   - **Operations:** Process Create
   - **Actions:** Tag Target
   - **Tags to Add/Remove:** childofsvchost
   - **Process:** svchost.exe
2. Create a second Custom Rule that reports creation of processes identified with the tag created in the previous rule.
   - **Operations**: Process Create
   - **Actions:** Report
   - **Process Tag(s):** childofsvchost

Example: Promote an Installer and Demote its Children

Perhaps you want to promote an installer so that it can successfully install an application, but you do not want the application to be able to create files that are automatically approved. For example, you might want to allow installation of notepad++, but not have scripts created by notepad++.exe be approved based on this promotion:

- **Operations:** Write
- **Actions:** Promote process, Don't Promote Children
- **Path or File:** *\notepad++.exe
- **Process:** Any Process

Switching to or from Expert Rule Mode

You can change an existing rule to an Expert Rule if you want to add operations or actions not available in its current rule type. When you make this change, the menu choices you began with should be converted to the correct checkboxes in Expert Mode.

Important: Avoid changing rules in the other direction, from an Expert Rule to a non-expert rule. Many of the operations and actions in Expert rules are not available to other rule types, and you could lose your rule definition entirely or convert the rule to something that does not match the operations or take the actions you want.

Contacting Carbon Black Support

For your convenience, support for Cb Protection is available through several channels:

Technical Support Contact Options
Web: User eXchange (login required)
E-mail: support@carbonblack.com
Phone: 877.248.9098
Fax: 617.393.7499

Reporting Problems

When you call or e-mail technical support, please provide the following information to the support representative:

Required Information	Description
Contact	Your name, company name, telephone number, and e-mail address
Product version	Product name (for example, Cb Protection Server or Agent) and version number
Hardware configuration	Hardware configuration of the server or endpoint having the issue (processor, memory, and RAM)
Document version	For documentation issues, specify the version of the manual you are using. The date and version of the document appear on the cover page of most documents and after the Copyrights and Notices section of longer manuals.
Problem	Action causing the problem, error message returned, and event log output (as appropriate)
Problem severity	Critical, serious, minor, or enhancement