



Application Control
powered by AppSense

Product Guide

Version 10.1 FR2

Copyright Notice

This document contains the confidential information and/or proprietary property of Ivanti, Inc. and its affiliates (referred to collectively as "Ivanti"), and may not be disclosed or copied without prior written consent of Ivanti.

Ivanti retains the right to make changes to this document or related product specifications and descriptions, at any time, without notice. Ivanti makes no warranty for the use of this document and assumes no responsibility for any errors that can appear in the document nor does it make a commitment to update the information contained herein. For the most current product information, please visit www.ivanti.com.

Copyright © 2003-2017, Ivanti. All rights reserved.

Ivanti and its logos are registered trademarks or trademarks of Ivanti, Inc. and its affiliates in the United States and/or other countries. Other brands and names may be claimed as the property of others.

Table of Contents

Product Guide	1
Copyright Notice	2
Table of Contents	3
Product Overview	6
Functionality	6
Features	7
Benefits	10
Product Architecture	11
Application Control Console	17
Application Control Security Methods	24
Licensing	32
Managing Licenses	32
Export License Files	34
Import License Files	34
Troubleshooting	34
Service Packs	35
Installing Service Packs	35
Rolling Back Service Packs	36
Configuration	37
Configuration Elements	37
Default Configurations	39
Maintain Configurations	43
Global Settings	48
Trusted Owners	48
Extension Filtering	52
Application Termination	53
Message Settings	56
Archiving	67
Policy Change Requests	72
Help Desk Portal	77
Manage	81
Advanced Settings	81
Signature Hashing	100
Auditing	104
Configuration Profiler	109
Configuration Change Tracking	110
Privilege Discovery Mode	114
Privilege Discovery Results	118
Group Management	123
Create a Group	123
Add Items to a Group	123
Add Groups to a Rule Item	125
Remove Groups from a Rule Item	126

Delete a Group	126
Capture Signatures in a Group	127
Rules	128
Security Levels	129
Policy Change Request Settings	131
Group Rules	132
User Rules	133
Device Rules	133
Custom Rules	134
Scripted Rules	135
Process Rules	140
Rule Options	144
Rules Items	168
Control Applications	178
Use Process Rules to Restrict Access to FTP	180
Rules Examples	181
Condition Management	187
Create a Condition	189
Reusing Conditions	190
Condition Variables	190
Field Validation	192
Computer Conditions	194
Scripted Rules	196
Directory Membership Conditions	201
Scripted Conditions	202
User Privileges	205
User Privileges Policies	205
Create a User Privilege Management Policy	206
Add Group Membership to a Policy	207
Assign Privileges to a Policy	207
Privileges	210
User Privilege Management	215
System Controls	231
Self-Elevation	238
Application Network Access Control	244
Network Connection Items	244
Add a Network Connection	244
Add a Network Item Directly to a Rule	246
Edit a Network Connection Directly in a Rule	246
Assign a Network Connection Item to a Group	246
Edit a Network Connection Item in a Group	247
Application Network Access Control and Reverse DNS Lookup	247
Configure Reverse DNS Lookup Entries	247
Endpoint Configuration Merging	248
Merge Components	248

ManifestGen Tool	249
Manifests	249
Merge Configurations	253
Merge Behaviors	255
Live Configuration Rules	256
Live Configuration Update Behavior	257
Endpoint Configuration Merging Auditing Events	257
Endpoint Analysis	258
Endpoint Analysis Preparation	259
Working with Endpoint Analysis	260
Installed Applications Scans	261
Application Usage Scans	262
Application Data	263
Export an Endpoint Analysis Data File	263
Add Files to Configurations	263
Rules Analyzer	265
Prerequisites	266
Set Up Logging for Rules Analyzer	267
Log Files	268
Rules Analyzer Tasks	270
Sample Scripting Reference	271
Sample Script: Create UPM Policies	313
Sample Script: Add User Privileges Component	317
Sample Script: Edit User Privileges Component	319
Configuration Object	321
Configuration Helper Object	350
Import and Export Scripted Rules	355
Appendix	357
Citrix XenApp	357
Web Services Configuration	358
Wildcards and Regular Expressions	371
Distributed File Systems	372
App-V5.0 Support	373
Components	374

Product Overview

Application Control prevents unauthorized code execution and enforces software licensing through the “trusted ownership” model and uses an improved approach to traditional whitelisting and blacklisting. It also manages user privileges and policy at a granular level whilst allowing for optional self-elevation when exceptions occur. Application Control keeps IT security requirements in balance with user productivity needs, delivering endpoint security through privilege and application control, increased corporate compliance, improved platform stability and consistency, and significant reductions in both IT support and software licensing costs.

Functionality

Application Control main feature set includes:

- Application Access Control
- Application Network Access Control
- Privilege Management

You can turn off any of these parts of functionality if they are not required. For example, you may not want to use Application Network Access Control.

To enable or disable certain Application Control functionality:

1. In the Manage ribbon, Click **Advanced Settings**.

The Policy Settings tab is displayed.

2. In the Functionality region, select to enable or deselect to disable one or more of the following Application Control functionalities:
 - Application Access Control
 - Application Network Access Control
 - Privilege Management

All the functionality options are selected by default.

3. Click **OK**.

Features

Application Control provides the following key features for application control:

Privilege Management

Privilege Management allows you to create reusable user privilege policies which can be associated with any rules and can elevate or restrict access to files, folders, drives, signatures, application groups, and Control Panel components. A more granular level of control allows you to assign specific privileges for debugging or installing software, or to set integrity levels for managing interoperability between different products, such as Microsoft Outlook and Microsoft Word.

Privileges Management contains four primary functions:

- Elevating user privileges for applications.
- Elevating user privileges for Control Panel components and Management Snapins.
- Reducing user privileges for applications.
- Reducing user privileges for Control Panel components and Management Snapins.

Trusted Ownership

By default, only application files owned by an administrator or the Local System are allowed to execute. Trusted Ownership is determined by reading the NTFS permissions of each file which attempts to run. Application Manager automatically blocks any file where ownership cannot be established, such as files located on non-NTFS drives, removable storage devices, or network locations. These files can optionally be allowed to run either by specifying them as Allowed Items or by configuring a Self-Authorizing User rule. The Trusted Owner list can be configured to suit each environment.

Rules: User, Group, Device, Custom, Scripted, and Process

Extend application accessibility by applying rules based on username, group membership, computer or connecting device, scripts and parent processes, or combinations of these. Allowed Items and Denied Items, Trusted Vendors, and Privilege Management can be specified in each rule, and are applied to a user session based on the environment in which the user operates.

Scripted Rules

Scripted rules allow administrators to apply Allowed Items, Denied Items, Trusted Vendors, and Privilege Management policies based on the outcome of a Windows PowerShell or VBScript. Scripts can be run for each individual user session or run once per computer.

Process Rules

Process rules apply to parent processes to manage access to child processes at the next level below the parent processes. Process rules include Allowed Items, Denied Items, Trusted Vendors and Privilege Management. The rule does not manage access to the parent process.

Trusted Vendors

Allow authentic applications to run when they have digital certificates signed by trusted sources, and are otherwise prohibited by Trusted Ownership checking. Define a list of Trusted Vendor certificates for each User, Group, Device, Custom, Scripted, and Process rule in the configuration.

Application Termination

Application Termination allows you to control triggers, behavior, and warning messages for terminating applications on managed computers. You can also control the manner in which applications are terminated and how the user is notified.

Network Connections

Block access to certain applications accessed via IP, Universal Naming Convention (UNC) or host name. Application Control can manage access based on the location of the requester, for example if they are connecting via a virtual private network (VPN) or directly to the network.

Digital Signatures

SHA-1, SHA-256 and Adler-32 signature checks may be applied to any number of application control rules, providing enhanced security where NTFS permissions are weak or non-existent, or for applications on non-NTFS formatted drives. A digital signature wizard allows easy creation and maintenance of large digital signature lists.

Windows Store Apps

Access to Windows Store apps can be controlled by Application Control. Grant or restrict access by applying group rules to one or more Windows Store apps. Application snippets can also be imported and rules configured if the machine being used to create the configuration is not compatible with Windows Store apps.

Endpoint Analysis

Allows an administrator to browse to any endpoint and retrieve a list of applications that have been installed on that endpoint. Search for any executable files and add them to the configuration. Application Control records which applications are started and by whom. The recording of data is started and stopped by the administrator. Organize the files into authorized and unauthorized groups to quickly create a policy. The configurations can be deployed to a user, a group of users, a machine, or a group of machines. Endpoint Analysis is on demand and inactive by default.

Offline Entitlement

Users are increasingly mobile. So it is important that entitlement rules are enforced when the user is not connected to the corporate network. Application Control ensures users only access the applications and resources to which they have permission when offline by using entitlement rules on the endpoint device.

Passive Monitoring

Application Control can monitor application use without preventing users running applications. Passive monitoring can be enabled or disabled on a per user, device, or group basis and provides a tool to track user behavior prior to full implementation, or to understand application use for software license management.

Self-Authorizing Users

Provides the option for users to execute applications that they have introduced into the system. Applications can be added to a secure machine while outside of the office without relying on IT support. A comprehensive audit can detail information such as the application name and the time and date of execution and device. Additionally, a copy of the application can be taken and stored centrally for examination.

Application Limits and Time Restrictions

Apply a policy to control the number of application instances a user can run, along with the times when it can run. You can create a policy to control or enforce licensing models by controlling application limits on a per user basis, but not per device.

Configuration Templates

Best practice configuration templates are provided that can be imported into Application Control. Application Control can import a number of configuration files and use these in combination.

Privilege Discovery Mode

Allows you to monitor endpoints to identify applications that use administrative rights. A web service is used to collect the data and relays that data to the Privilege Discovery Mode work area in the Application Control Console.

Auditing

Events are raised by Application Control according to the default Event Filtering configuration and audited directly to a local file log or the Windows Event Log. Alternatively, events can be forwarded for auditing to the Management Center via the Deployment Agent (CCA). The Application Control audit event reports available in the Management Center can also be used to provide details of current application usage across the enterprise.

Windows Scripting Host Validation

The default configuration in Application Control validates all Windows Scripting Host (WSH) scripts, such as VBS, against configuration rules. This ensures that users can only invoke authorized scripts, eliminating the risk of introducing WSH scripts that contain viruses or malicious code.

The Validation settings can be disabled in the Application Control **Options** dialog, along with validation of cmd.exe, self-extracting ZIP files, registry files, and Windows installer (MSI) files.



Only self-extracting EXEs formatted using the ZIP specification are supported. For additional information, see [ZIP Specifications](#).

Functionality Cut-off Settings

Enable and disable certain features in Application Control either when not in use or when troubleshooting issues in your configurations. The functionality that you can manage in this way includes:

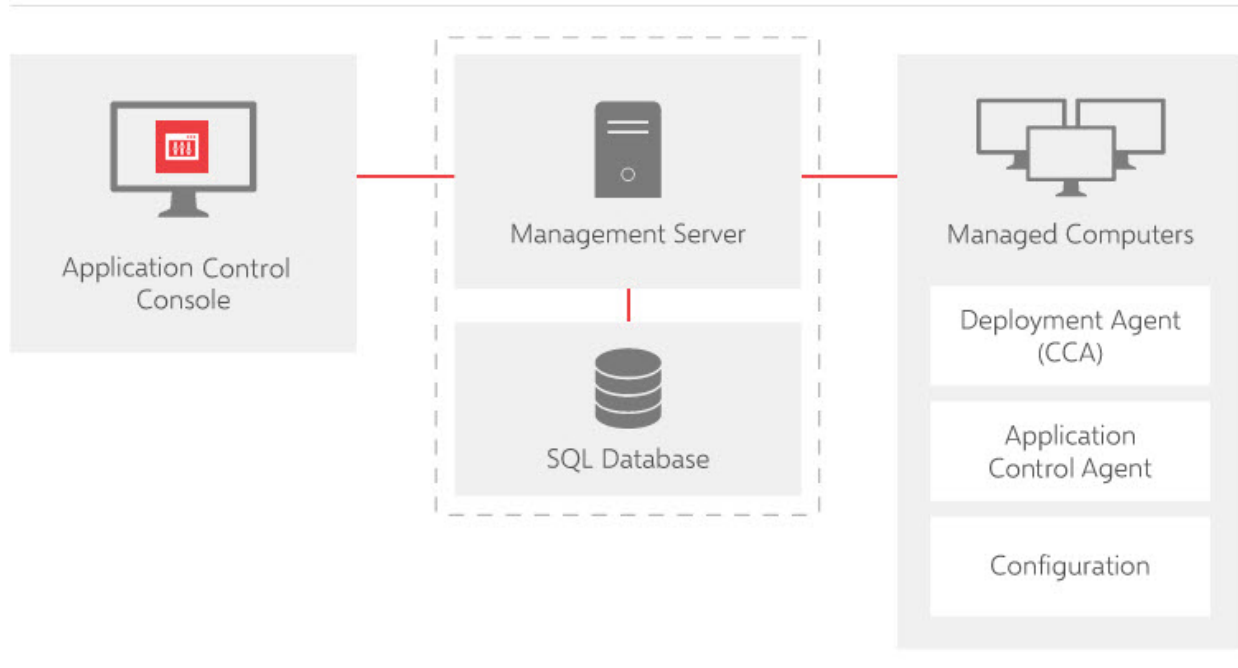
- Application Access Control
- Application Network Access Control
- Privilege Management

Benefits

The key benefits of using Application Control are:

- Protect against malicious code.
- Selectively elevate or restrict administrative privileges to access or run specific applications or access system settings.
- Control role-based application usage.
- Protect 'out of the box' against all unauthorized application usage.
- Manage processes at a granular level to control application access to child processes.
- Stop unauthorized device license usage.
- Apply time restrictions on when applications can or cannot be run.
- Control network access from within applications.
- Control network access based on location.

Product Architecture



Software Agent

Application Control is installed and run on endpoints using a lightweight agent. The agent is installed directly onto the local computer. Both agents and configurations are constructed as Windows Installer (MSI) packages and so can be distributed using any third party deployment system that supports the MSI format. The installers are delivered in separate 32-bit and 64-bit Microsoft Installer packages.

For Application Control to function, the agent must be installed on the client endpoint together with an associated configuration. The installation may be performed manually or by means of a deployment system such as the Management Center. Because agents and configurations are installed and stored locally on the endpoint, they continue to operate when the endpoint is disconnected or offline.

The Application Control agent installs a Windows Service (the Application Control Service), a filter driver, and a hook. The hook sits above the driver and intercepts all executables. It does not intercept DLLs, unlike the driver. If an executable is not intercepted by the hook it is intercepted by the driver.

The driver intercepts execution requests that are made within the operating system that pass from the I/O Manager to the drive and the device subsystems, for example NTF5.SYS or the LanMan Redirector for Microsoft Networking Services. The driver does not intercept ordinary file access such as the opening of a document or text file.

Every intercepted create process request is intercepted by the hook. When the request is intercepted by the hook, the request is passed on to the Application Control Agent Service for validation against the configuration settings, which returns an execution granted or denied response that is dealt with by the hook or driver, depending on which sent the request. If the response is granted, the request is passed on to the relevant file system driver to continue with the application loading from disk.

In the case of a denied executable or script, the agent replaces the original path with Application Control's customizable message box (AMMessage). This effectively blocks access to the original requested executable and instead displays a message to the user. In the event of a DLL being blocked, no message is displayed and the default operating system message is displayed.

Agent Service

The Application Control Agent Service runs as a SYSTEM service on each computer that is controlled using the Application Control component. The agent provides the intelligence for dealing with the execution requests passed from the Application Control kernel level driver and the hook. Each and every execution request is validated against the configuration settings that are held on each local machine containing the Application Control agent software. Along with the details of the application request, the agent service checks who the user is and which computer the request originates from so that this can be processed at the same time to enable user, group, client, or custom rules to function as expected.

The configuration is stored in a local configuration file for performance and control reasons. This means that all requests can be turned around in minimum time and perhaps more importantly, without the need for a network link to a central server, which ensures that unconnected machines, such as laptops, remain secured even when not physically connected to the Local Area Network.

Agent Assist

Agent Assist provides support for the agent. Instances of Agent Assist are started on demand by the agent and run using the SYSTEM account. Each Agent Assist is specific to a user session. If Agent Assist is initiated, no more than one instance runs in a session. Once started, Agent Assist typically remains running until the session logs off or the agent is stopped.

Agent Assist does the following:

- Enforces time limits on applications.
- Prompts Self Authorizing Users to confirm whether to allow denied DLLs (applications are handled by Agent Assist).
- Performs auditing for the events, 9006, 9007, 9017.
 - 9006 - Self-authorization decision by user.
 - 9007 - Self-authorized execution request.
 - 9017 - An application has been terminated by Application Control.
- On 64-bit systems, Agent Assist can start the 32-bit DLL component that installs the 32-bit Application Hook into 32-bit applications running in the same user session.

DLL Injection Assist

DLL Injection Assist is a 32-bit component that is only installed on 64-bit systems. It is used solely by Agent Assist to install the 32-bit application hook into 32-bit applications running in the same user session.

Filter Drivers

The agent intercepts, then validates, all application execution requests against the configuration. It then either grants or denies access to the executable content. The agent also triggers auditing events that are collected by the AppSense Deployment Agent.

The driver only intercepts execution requests placed against the operating system because it is connected between the I/O Manager (in the Executive Services) and the actual device drivers for the file systems themselves (for example, NTFS.SYS, CDROM.SYS, or LanMan Redirector for Microsoft Networking Services). The driver does not intercept ordinary file access such as the opening of a text file, document, or presentation.

Every intercepted request is subsequently passed on to the Application Management Agent Service for validation against the current configuration. The Agent Service returns an allowed or denied response that is dealt with by the filter driver. If the response is allowed, the request is passed on to the relevant file system driver to continue with the application loading from disk. If the request is denied, the filter driver replaces the request with Application Control's error-handling system, which is responsible for the display of a fully customized message box to the end user. This error handling effectively blocks access to the requested executable code by advising the originating process that all is successful, and the customized message box is displayed in place of the expected executable code. This prevents the operating system displaying a 'File not Found' or 'Access Denied' message.

The driver is a lightweight driver that filters file system requests for files, but not folders, with the Execute, Overwrite, and Rename permissions requests. The driver sends requests to the Application Control agent for authorization. Depending on the response from the agent, the driver allows, redirects, or denies the request.

When it redirects, the driver redirects to one of the Message Box applications. The driver only redirects as a fall back if the request is missed by the hook.

The filter driver can dynamically start but cannot be stopped without a reboot. This can be found in %systemdrive%\ProgramFiles\ApplicationManager\Agent\AmFilterInstall and is called

For more information on the Deployment Agent, see the [Management Center Help](#).

Mini Filter Driver

The mini filter driver is a lightweight driver that filters file system requests for both files and folders on UNC paths, but not for local drives. The driver sends requests to the agent for authorization. Depending on the response from the agent, the driver allows or denies the request.

This can be found in %systemdrive%\Program

The mini filter driver can be dynamically started and stopped.

Application Hook

This is a DLL that is loaded into every user process.


The Application Hook sends create process and network requests to the agent for authorization. In the event of a blocked executable, the original request is replaced with a request for AMMessage. In the event of a blocked network request, access to the network resource is denied. If any token modification is required, as part of Privileges Management, an appropriate request is sent to the agent. The agent sends back a modified token, which is used to launch the requested process.

Where Application Network Access Control (ANAC) is concerned, because requests for network traffic is high, the results provided by the agent are cached in the memory of the application. This is essential to avoid a dramatic performance degradation to network traffic.

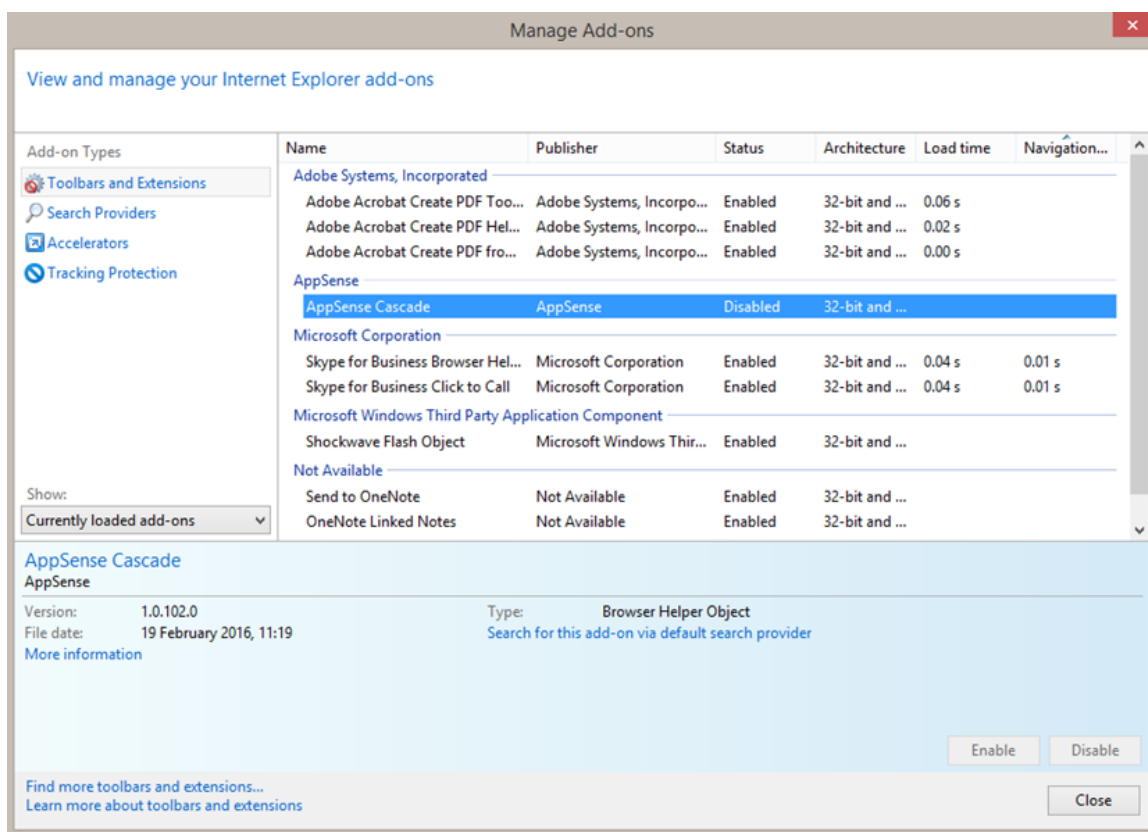
For more information on ANAC see [Application Network Access Control](#).

Browser Add-on

CascadeBHO.dll is an Application Control Browser Helper Object (BHO) loaded by Internet Explorer that is used as part of the URL Redirection and Elevated Web Sites features. If a configuration contains any of these types of rules, the Cascade BHO is enabled, and therefore loaded, by Internet Explorer. If there are no URL Redirection or Elevated Web Sites rules, the BHO is disabled.

 The BHO is loaded by only Internet Explorer. A separate extension that provides the same functionality, AppSense Cascade, is loaded by Chrome. There is no equivalent for the Microsoft Edge browser.

Check the CascadeBHO.dll Status



1. In Internet Explorer, select **Tools > Manage add-ons**.

The Manage Add-ons dialog displays.

2. In the Add-on Types panel, ensure **Toolbars and Extensions** is selected.

You can view the status of the add-ons in the pane on the right.

Configuration

AppSense Application Control configuration files (AAMP files) contain the rule settings for securing your system. The agent checks the configuration rules to determine the action to take when intercepting file execution requests.

Configurations are stored locally in the All Users profile and are protected by NTFS security. In standalone mode, configuration changes are written directly to the file system from the Application Control console. In Enterprise mode, configurations are stored in the Management Center database, and distributed in MSI format using the Management Center console.

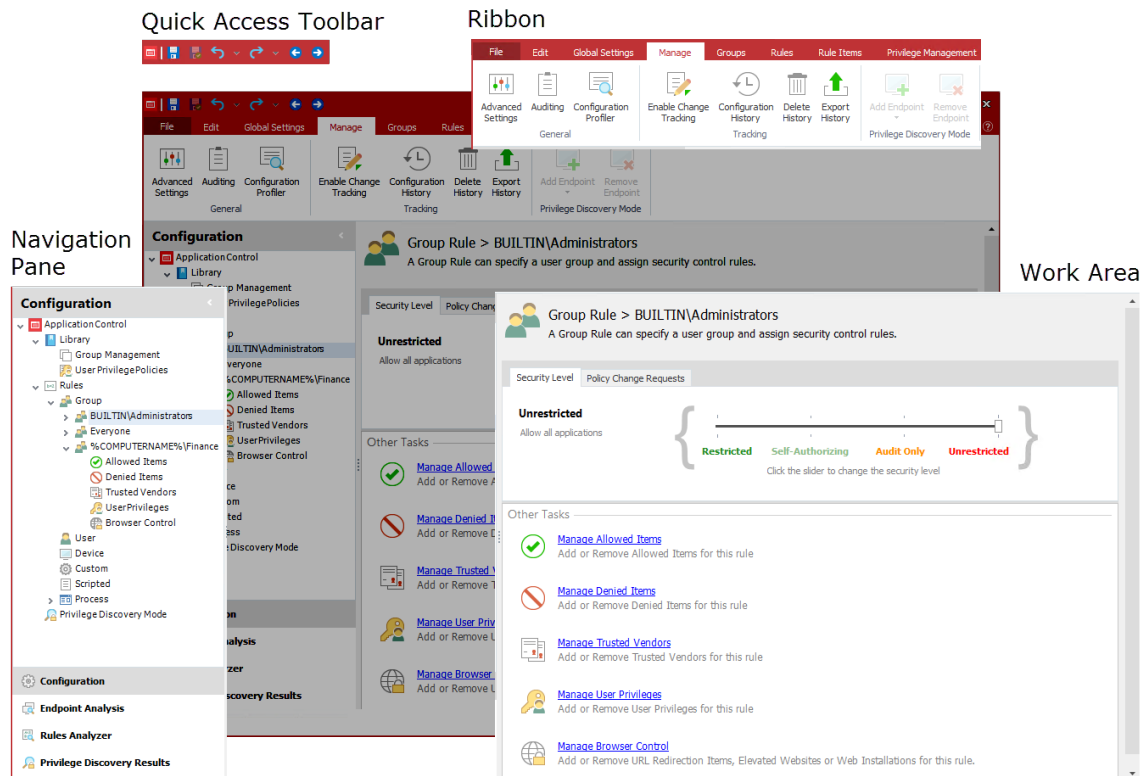
Configurations can also be exported and imported to and from MSI file format using the Application Control console. This is useful for creating templates or distributing configurations using third party deployment systems.

After creating or modifying a configuration you must save the configuration (and deploy if necessary) to ensure that they are actioned.

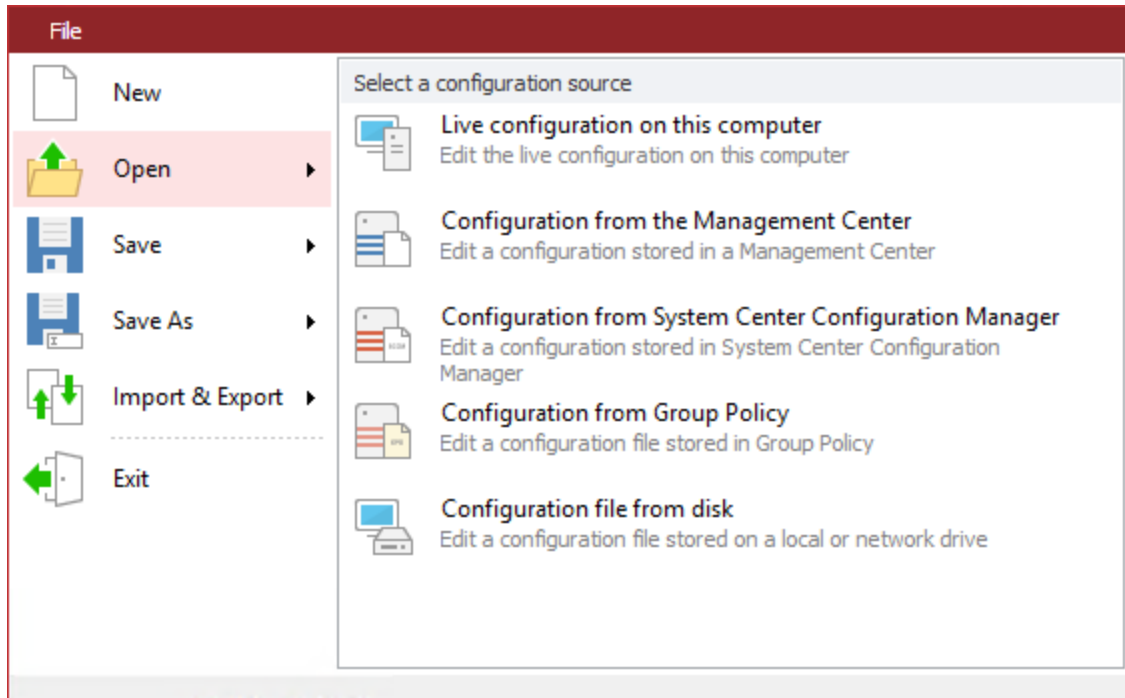
Application Control Console

The console enables you to create, view, edit, and save configurations for Application Control. The console includes the Configuration Profiler, which you can use to review the probable effect of the configuration on users. The Rules Analyzer function allows you to record the actual effect of the configuration on users on an endpoint that has the Application Control agent installed and running. The Endpoint Analysis tool allows you to record application usage, and to catalog installed application usage on an endpoint that has the Application Control agent installed.

Console Elements



File Tab Application Menu



The Application menu provides options for managing configurations, including create new, open existing, save, and to import and export configurations.

Option	Description
New	Creates a new default configuration which is locked for editing.
Open	<p>Opens an existing configuration (AAMP format) from one of the following locations:</p> <ul style="list-style-type: none"> • Live configuration on this computer • Configuration from the Management Center • Configuration file on a local or network drive • Configuration from System Center Configuration Manager • Configuration from Group Policy <p>A live configuration is located on a computer that has an Application Control agent installed and running.</p>
Save	<p>Saves the configuration in one of the following states:</p> <ul style="list-style-type: none"> • Save and continue editing - saves the configuration and keeps it locked while open for editing. Any changes that have been made are not committed to the configuration and it cannot be deployed while locked.







Option	Description
	<ul style="list-style-type: none">• Save and unlock - saves the configuration and unlocks it ready for deployment. The current configuration closes and a new default configuration opens.• Unlock without saving - unlocks the configuration without saving changes. The current configuration closes and a new default configuration opens.
Save As	<p>Saves the configuration with a new name to one of the following locations:</p> <ul style="list-style-type: none">• Live configuration on this computer• Configuration in the Management Center• Configuration file on a local or network drive• Configuration in System Center Configuration Manager - Saves your configuration to the specified System Center Configuration Manager server.• Configuration in Group Policy - Creates the configuration in a selected Group Policy Store. <p>A live configuration is located on a computer that has an Application Control agent installed and running. If using a Microsoft Windows operating system with UAC enabled, you must ensure that you open the console with administrator privileges.</p>
Import & Export	<ul style="list-style-type: none">• Imports a configuration from MSI format, usually legacy configurations which have been exported and saved from legacy consoles.• Exports a configuration to MSI format.
Exit	Closes the console. You are prompted to save any changes you have made to the current configuration.
Preferences	Allows you to choose whether to show the splash screen on startup.

Quick Access Toolbar



The Quick Access toolbar provides quick functionality for managing the configuration setup, such as Save, Save and Unlock, Undo, Redo, and navigation to previously and next displayed views.

Quick Access Toolbar Options

Option	Description
	Save Saves changes to the configuration. The configuration will remain locked if opened from the Management Center.
	Save and unlock Saves changes and unlocks the configuration. These changes can now be deployed from the Management Center.
	Undo Clears the action history. Up to 20 previous actions are listed. Select the point at which you want to clear the actions. The action selected and all proceeding actions are undone.
	Redo Re-applies the cleared action history. Up to 20 cleared actions are listed. Select the point at which you want to redo the actions. The action selected and all subsequent actions are redone.
	Back Navigates back through the views visited in this session.
	Forward Navigate forward through the views visited this session.

Manage the Quick Access Toolbar

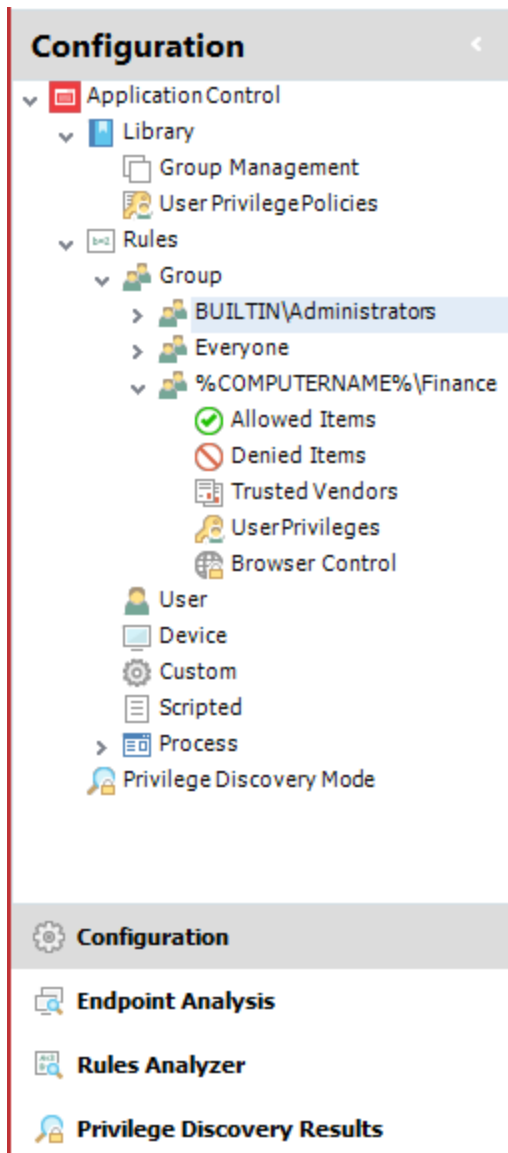
You can configure the Quick Access toolbar to display the commands you use the most and to change its position in the console:

- To add a command to the Quick Access Toolbar, right-click the ribbon button or file menu option and select **Add to Quick Access Toolbar**.
- To remove a toolbar item, right-click it and select **Remove From Quick Access Toolbar**.
- To display the toolbar below a ribbon, right-click a ribbon or the toolbar and select **Show Quick Access Toolbar Below the Ribbon**.

Help

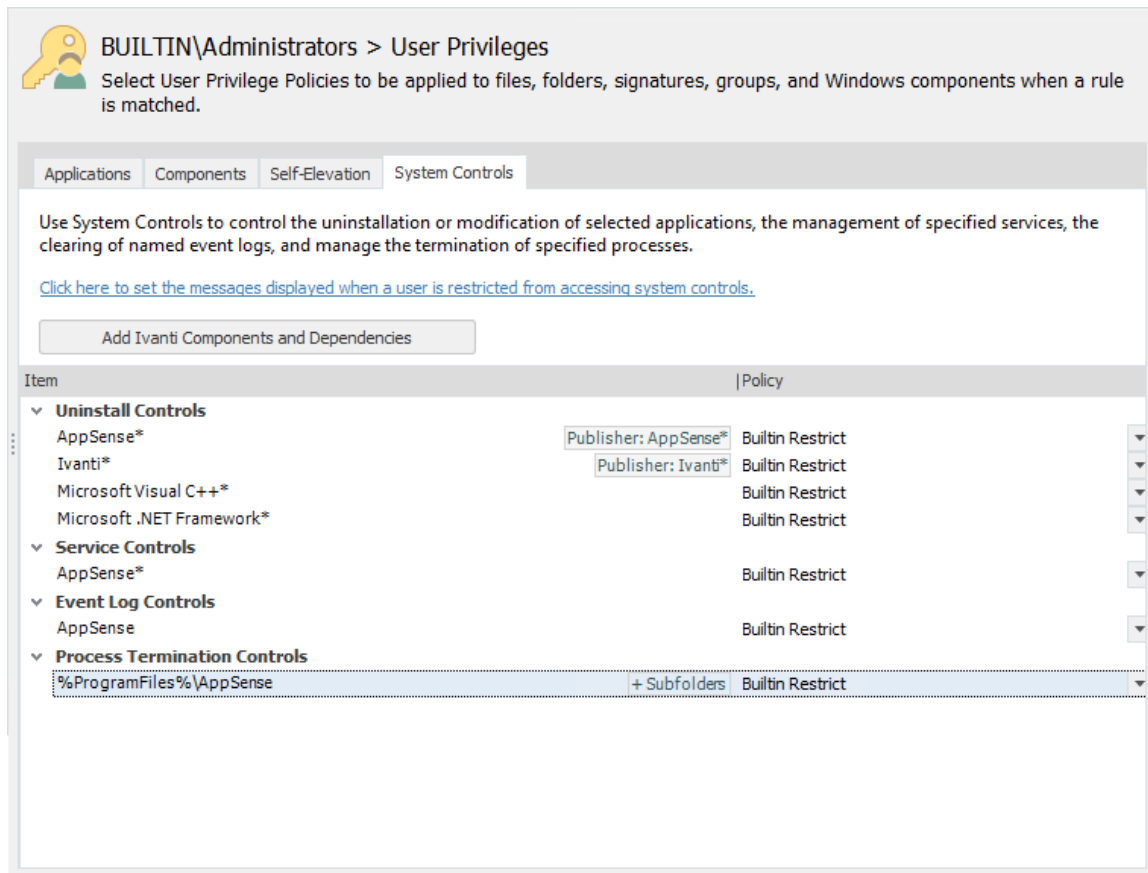
The **Help** ribbon includes a **Help** button that launches the Help for the product and displays the topic relating to the current area of the console in view. A smaller icon for launching the Help displays at the far right of the console, level with the ribbon tabs, for convenience when the **Home** ribbon is not in view. You can also press **F1** to launch the Help topic for the current view. The ribbon also contains the **About** button, which you click to display the product version and build number, and buttons to visit the Ivanti website and to contact Support.

Navigation Pane



The **Navigation** pane consists of the navigation tree and navigation buttons. The navigation tree is the area for managing nodes of the configuration. The navigation buttons allow you to view the different areas of the console.

Work Area



The **Work Area** provides the main area for managing the settings of the configuration and product. The contents of the work area vary according to the selected nodes in the navigation tree and the selected navigation buttons. Sometimes the work area is split into two panes. For example, one pane can provide a summary of the settings in the other pane.

Additional Console Features:

- **Shortcut Menu** — right-click shortcuts are available in the navigation tree and some areas of the console.
- **Drag and Drop** — this feature is available in some branches of the navigation tree.
- **Cut/Copy/Paste** — these actions can be performed using the buttons in the Edit ribbon, shortcut menu options, and also using keyboard shortcuts.
- Recommended screen resolution for the console is 1024 x 768 pixels.

Application Control Security Methods

Application Control offers a number of security methods that you can implement to protect a system without complex lists and constant management. These include the following:

- Trusted Ownership
- Trusted Vendors
- Digital Signatures
- Whitelisting
- Blacklisting

To get the most value out of an Application Control configuration, you can use hybrid approach in which you combine the most suitable components from each security method to provide the optimum security model, while minimizing overall management and configuration overheads.

The Trusted Ownership approach enables new applications to be installed by Trusted Owners without any changes required to the Application Control configuration, yet still provides full security against unknown application and script content introduced by non-trusted end users. So it's recommended that this security method be used for the basis of most Application Control configurations. This is why this functionality is enabled by default in all new Application Control configurations.

The whitelist approach is the most secure but it is an administrative-intensive security model. If an enterprise does not use NTFS security on their file systems, the whitelist method is the recommended option because Trusted Ownership relies on the file owner information that is only found in NTFS.

Trusted Ownership is only appropriate for locally installed executable content; that is, applications that exist on local fixed drives in a computer. Any executable or script content that resides on network locations or on removable media, such as a CD or a DVDROM, is automatically considered untrusted, and is immediately blocked from executing. Any such application that must be executed by a user must be specifically added to the whitelist in the Application Control configuration, with a full UNC path to the relevant executable. It is possible to optionally disable Trusted Ownership checking on these items if necessary, or to optionally select to take a SHA-1, SHA-256 or Alder-32 signature to check the file at runtime. It is considered good practice to use digital signature checking for applications based on networks or removable media because these files tend to be outside of the control of the administrator responsible for the organization's endpoints.

Trusted Vendor checking is recommended for development and test environments where end users may need to constantly install and test different versions of company-owned application and script content. By signing the desired executables with a digital certificate, Trusted Vendor checking can be configured to allow all signed components to be executed as and when needed.

Finally, you should create a blacklist, preventing specific user access to applications that would typically be installed and therefore owned by Trusted Owners, including parts of the operating system such as registry editing tools, file sharing tools, and access to Control Panel components. This blacklist can also be used for application license management, when used in conjunction with whitelists and the Application Limits functionality.

Trusted Ownership

Application Control uses secure filter drivers and Microsoft NTFS security policies to intercept all execution requests. Execution requests go through the Application Control hook and any unwanted applications are blocked. Application entitlement is based on the ownership of the application, with default trusted ownership typically being for administrators. By using this method, current application access policy is enforced without the need for scripting or list management. This is called Trusted Ownership. In addition to executable files, Application Control also manages entitlement to application content such as VBScripts, batch files, MSI packages, and registry configuration files.



Application Control only supports PowerShell from version 2.0 inward and must be installed in the endpoint.

Trusted Ownership is the default method of controlling access to applications in Application Control. It uses the Discretionary Access Control (DAC) model. It examines the owner attribute of the file and compares it to a predefined list of trusted owners. If the owner of the file appears in the list then execution of the file is granted, otherwise it is denied. The decision is made independently of the user actually trying to execute the file.

An important feature of this security method is the ability to not consider the file contents itself. In this way, Application Control is able to control both known and unknown applications. Conventional security systems such as anti-virus applications compare file patterns against those in a known list to identify potential threats. So the protection it offers is directly proportional to the accuracy of the list that it uses for comparison. Many malware applications are either never identified, or at best, identified only after a period of time while systems are left vulnerable. Application Control, by default, allows ALL locally installed executable content to execute IF the owner of the executable is listed in the Trusted Owners list in the configuration. The administrator must then supply a list of applications that they do not want to execute from the local disk subsystem, which would typically be administrative applications such as mmc.exe, eventvwr.exe, setup.exe, and so on.

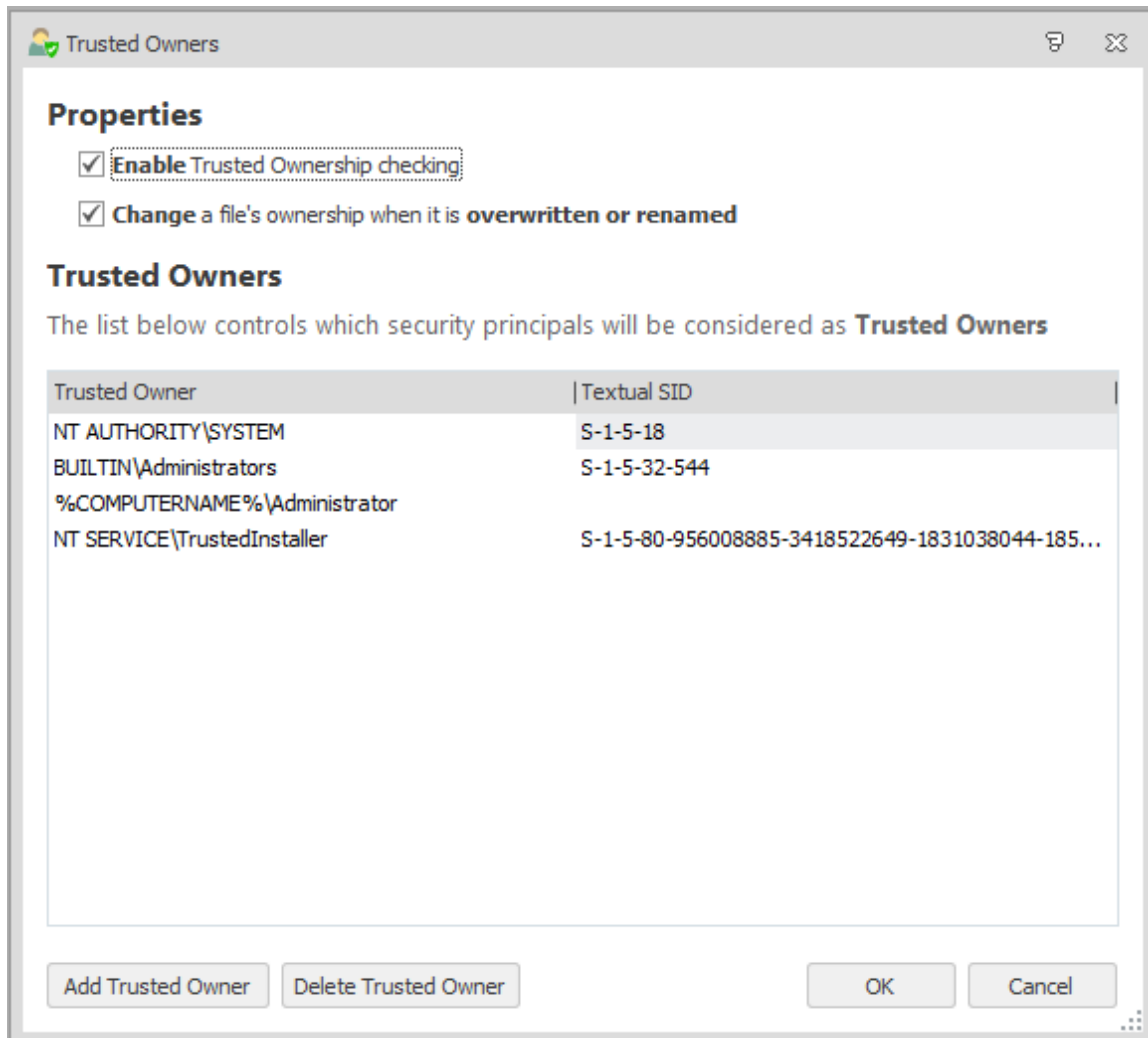
If this approach is taken, the administrator does not have to find out the full details of every piece of executing code required for the application set to function because the Trusted Ownership model allows / denies access as appropriate.



Although Application Control is able to stop any executable script based malware as soon as it is introduced to a system, Application Control is not intended to be a replacement for existing malware removal tools, but should act as a complementary technology sitting alongside them. For example, although Application Control is able to stop the execution of a virus, it is not able to clean it off the disk.

Application Control and Trusted Ownership

Application Control maintains a trusted owners list that is defined in the Trusted Owners dialog. This dialog is accessed from the Global Settings ribbon.



Users and groups can be deleted or added as required.



Do not remove all Trusted Owners. This would result in no application on the system being trusted and standard users unable to run anything.

In the NTFS system, a file can be owned by either a user or a group and therefore both may be added. When the check for Trusted Ownership is performed the System Identifier (SID) of the file owner is determined and this is checked against the list of SIDs in the trusted owner configuration. Application Control does not evaluate a group or determine users of a group. This ensures that Application Control continues to function correctly when machines are not connected to a network and this information is not available.

There are two options in the Trusted Owners dialog:

- **Enable Trusted Ownership checking** - Select to switch on Trusted Ownership checking. If this is not selected Application Control does not perform any Trusted Ownership checking and other security methods must be configured to give the desired security.
- **Change a file's ownership when it is overwritten or renamed** - The default for certain operating systems is to retain file ownership when a file is overwritten or renamed. This can be seen as a security flaw as if NTFS permissions allow, a user may overwrite a legitimate file with a file that would otherwise be blocked. Select this option to ensure that if a legitimate file is compromised in this way, the ownership changes to that of the user and Trusted Ownership prevents the file from being executed.

Trusted Ownership Rule

Trusted Ownership does not need to take into account the logged-on user. It does not matter whether the logged-on user is a Trusted Owner, administrator, or not. Trusted Ownership revolves around which user (or group) owns a file on the disk. This is typically the user who created the file.

It is common to see the group BUILTIN\Administrators in the Application Control console as the file owner. It is also possible to find that the file owner is an individual administrator's account. This results in the following situations:

- The file owner is the group BUILTINAdministrators and this group is a Trusted Owner. Trusted Ownership allows the file to execute.
- The file owner is an individual administrator and the individual administrator is a Trusted Owner. Trusted Ownership allows the file to execute.
- The file owner is an individual administrator and the individual administrator is not a Trusted Owner, but the BUILTIN/Administrators group is a Trusted Owner. Trusted Ownership does not allow the file to execute.

In the last case, even though the administrator who owns the file is in the BUILTIN/Administrators group, the file owner is not trusted. The group is not expanded to find out whether the individual owner should be trusted. In this case, to allow the file to execute, the file's ownership must be changed to that of the BUILTIN/Administrators.

Trusted Vendors

Trusted Vendors can be specified in each Application Control rule node. Trusted Vendors are used for listing valid digital certificates. A digital certificate is an electronic document that uses a digital signature to bind together a public key with an identity. This includes information such as the name of a person or organization, address, and so on. Digital certificates are issued by a certificate authority and used to verify that a public key belongs to an individual. Application Control queries each file execution to detect the presence of a digital certificate. If the file has a valid digital certificate and the signer matches an entry in the Trusted Vendor list, the file is allowed to run, and overrides any Trusted Ownership checking.

You can check whether a file has a digital certificate by displaying the Properties dialog. A file has a digital certificate if there is a Digital Signatures tab in which you can view details of the certificate including, signer information, advanced settings and an option to display the certificate.[trusted_Vendors.htm](#)

For more information, see [Add a Certificate to a Trusted Vendor](#)

Digital Signatures

Digital Signatures provide a means to accurately identify a file according to the actual contents of the file itself. Each file is examined and according to its contents, a digital hash, which may be likened to a fingerprint, is produced. Application Control makes use of the industry standard SHA-1, SHA256 and Adler-32 hashes. If the file is altered in any way, then the SHA-1 hash is also altered.



Other algorithms can be selected from the Signatures drop-down on the Advanced Settings dialog.

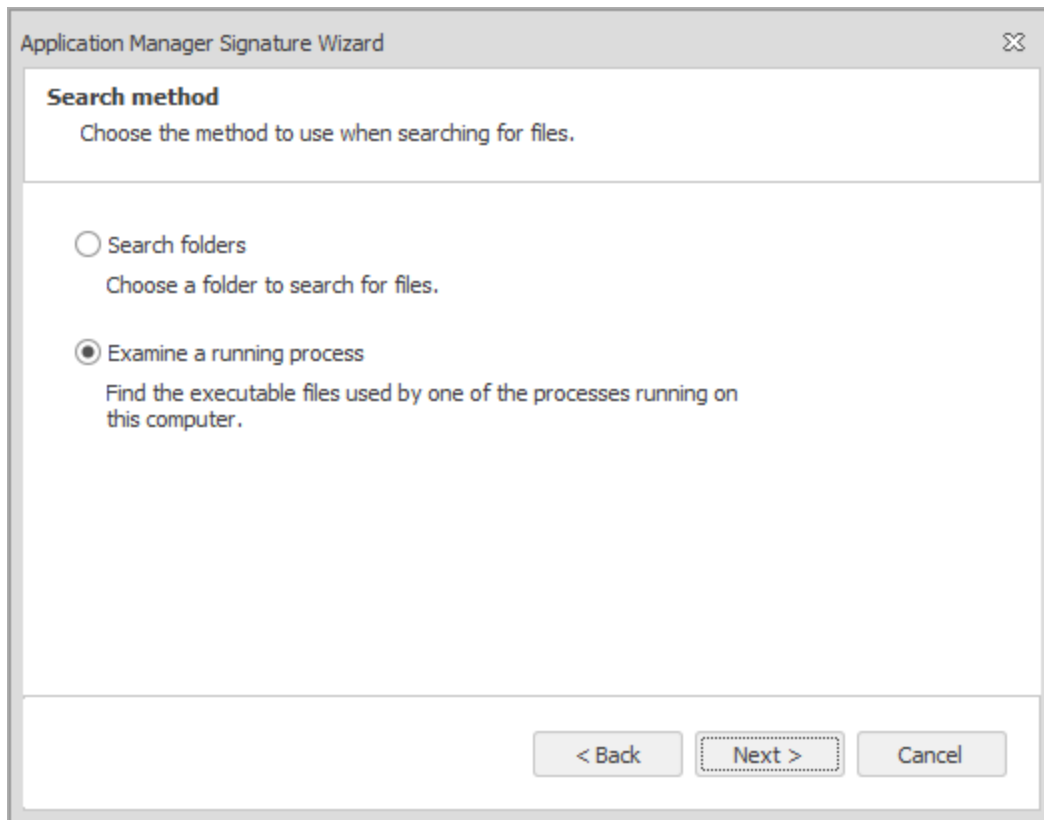
Digital hashing is seen as the ultimate security method because it is accurate. It identifies each file independently of all other factors other than the file itself. For example, an administrator takes a digital hash of all executables on a computer system and records them. A user then tries to execute an application. The digital hash of the application is calculated and then compared to the recorded values. If there is a match the application is granted execution, otherwise it is denied. This methodology also provides zero-day protection because not only does it stop new applications from being introduced, it also blocks any applications that have been infected with malware.

Although digital signatures provide a similar protection to Trusted Ownership, you must also consider the time and management involved with respect to maintaining the security systems in place. Applications are constantly being updated with service packs, bug fixes, and vulnerability patches. This means that all associated files are also constantly being updated. So if, for example, a service pack is applied to Microsoft Office then for the updated parts to work new digital hashes of the updated files must now be taken. Take care to ensure that these are available when the update is available to eliminate downtime. Additionally, it is recommended that you remove the old signatures.

Signature Wizard

Application Control has a Signature Wizard that allows you to apply digital signatures either to an individual file or a group. Digital signatures can be grouped in one of two ways, by means of scanning folders and subfolders, or by examining a running process.

The Signature Wizard is available from the Groups ribbon when you select a group beneath the Library > Group Management node.

The screenshot shows a dialog box titled "Application Manager Signature Wizard". It has a close button in the top right corner. The main area is titled "Search method" with the instruction "Choose the method to use when searching for files." There are two radio button options: "Search folders" with the subtext "Choose a folder to search for files." and "Examine a running process" with the subtext "Find the executable files used by one of the processes running on this computer." The "Examine a running process" option is selected. At the bottom, there are three buttons: "< Back", "Next >" (which is highlighted with a dashed border), and "Cancel".

The **Search Folders** option in the Signature Wizard scans all executable and script based files in selected folder and automatically calculates the digital hashes. The **Examine a running process** option allows you to select a process that is currently running. The process, along with all executable files it has currently loaded, is scanned and digital hashes calculated.

If a file is found for which the signature has already been calculated a notification of a duplicate is displayed. There is no need for a duplicate hash in a configuration. If the files are updated by means of, for example, a service pack, you can select the signature file group and choose to re-scan. All of the digital signatures are automatically updated and the new configuration can be deployed.

Whitelists

The whitelist approach dictates that every single piece of executable content must be predefined prior to the user making the request for the application on the operating system. Details of all the content identified in this way is kept on a whitelist that must be checked each time an execution request occurs. If the executable file is on the whitelist it is permitted; otherwise it is denied.

A small number of security technologies work in this way, but they often experience issues with the level of administration required once implemented. This is due to the necessity of adding and maintaining all patches, service packs, and upgrades to the whitelist.

Application Control fully supports this model of control, and adds significant steps to enable additional security in the model. One such addition is the ability to include SHA-1, SHA-256, and Adler-32 digital signatures, so that not only must the application name and file path match up, but so must the digital signature of that executable to that of a signature in the database. Furthermore, Application Control also adds the full path of the executable to the list to ensure that all three items match prior to application execution:

Filename - for example, winword.exe.

File Path - for example, C:\Program Files\Microsoft Office\Office\digital signature

To take the technology into the next stage of control, Application Control does not only take the details of the executables, but also requests that the administrator specify specific DLLs as well as all other executable content such as ActiveX controls, Visual Basic Scripts, and Command Scripts.

In Application Control, whitelists are as Allowed Items. Items in the Allowed Items list include:

- Files
- Folders
- Drives
- Signature Items
- Network Connection Items
- Windows Store Apps
- Groups
- Trusted Ownership
- Access Times

For more information, see [Allowed Items](#) and [Rule Items](#).

Blacklists

In contrast to whitelists, blacklists are a potential low security measure. A list is generated and then maintained that contains the applications that are to be denied execution. This is the main failing of this method, as it presumes that all dangerous applications are actually known about. This is of little use in most enterprises, specifically with email and internet access and / or where the user can introduce files and applications without administrator intervention.

Application Control does not need to actively maintain a list of denied applications because any applications not installed, and therefore owned by the administrator, are denied by use of Trusted Ownership.

One of the main reasons prohibiting applications via a blacklist is to enable Trusted Ownership to be used for license management by not allowing even known (and therefore trusted and owned) applications to run, until the administrator can later explicitly allow access to that very same application by defining a certain user / group or client rule. This protection needs no configuration, except to allow an outside application. Additionally, a blacklist is useful for denying access to files owned by trusted owners that can be deemed security risks. For example, regedit.exe, ftp.exe, and so on.

Licensing

The Licensing console allows you to manage User Workspace Manager product licenses.

The Licensing console allows you to:

- Manage licenses for single products, the User Workspace Manager suite and Evaluation licenses.
- Export license packages to MSI or LIC file format for saving to the Management Center or other computers which can be remotely accessed.
- Import and manage licenses from LIC file format.

For information about license deployment to endpoints, see [Management Center Help](#).

Managing Licenses

License details are included in the License Agreement which is issued when an order for our software has been completed.

The License Agreement includes the following information:

- Product, Feature, and Version Details
- Issue Date
- Expiry Date
- Customer Name
- Serial ID

Together with the license agreement you will receive either a TXT file or a LIC file. Use these in the Licensing Console to add or import the license.

Add a License

1. Open the Licensing console.
2. Click **Add**.

The Add License Key dialog displays.

3. Enter the License Key and click **Add**.

If you received a TXT license file, open the file and copy the license key, paste it in to the Add License Key dialog.

If you received a LIC license file, refer to "Import License Files" on page 34.

Details of the license are displayed in the console and the license key is added to %ALLUSERSPROFILE%\AppSense\Licenses

Activate a License

Once added, some licenses require activating.

1. Select a license or add one to the licensing console.
2. Click **Activate**.
3. Type or copy and paste the activation code.
4. Press **Enter** to accept the code.

The license console saves the license key to the MS Windows registry on the local machine. The License Status field updates to show the status of the license and the license details display in the lower part of the console.



To check that the license is active on your endpoint, search the registry for the license code. If the search finds the code, then the license is active.

Remove a License

1. Highlight the required license and click **Remove**.
A confirmation dialog displays.
2. Click **Yes** to confirm.

The selected license is deleted and removed from the console and the MS Windows registry or %ALLUSERSPROFILE%\AppSense\Licenses location, whichever is applicable to the license type.

Export License Files

Export licenses to an MSI or LIC file to create a backup and enable distribution to other endpoints using the Licensing console or the Management Center.

1. Highlight the license you want to export.
2. Click **Export** to display Windows Save As dialog.
3. Browse to the required location to save the license file.
4. Enter a name for the file.
5. Select the file type: MSI or LIC.
6. Click **Save**.

A file is created and saved in the selected location. This file can be copied to any network location and loaded via the Licensing console or in the Management Center console.

Import License Files

Import a previously exported license to an endpoint using the Licensing console.

1. Open the Licensing console.
2. Click **Import** to display the Windows Open dialog.
3. Navigate to the required LIC file.
4. Click **Open**.

Details of the license are displayed in the console and the license key is added to the following location:

%ALLUSERSPROFILE%\AppSense\Licenses

Troubleshooting

I received a license, what do I do?

If you have received a product license you can load the license by launching the Licensing Console on your client computer and entering the license code.

I have entered a license, but it says it is not activated, why?

Some licenses require activation before they can be used. Activation codes are provided by Ivanti. Activate a license by entering the License and Activation codes into the console.

Service Packs

Service Packs are self-contained packages or patches that are used to update specific files within a User Workspace Manager application without reinstalling the full application. Service packs can be applied more often and reduce the need for system restarts on your endpoints. Service packs are delivered as a Windows Installer patch (MSP) file and are often referred to as patch files.

Installing Service Packs

Service Packs can be installed or deployed using the same technology and techniques used when installing MSIs. Both Microsoft System Center and the Management Center 8 FR4 can deploy MSPs. If neither of these products are available, service packs can be installed using the command line interface.

For example, the command:

```
msiexec.exe /p ApplicationManagerAgent64.msp
```

installs any files that have been amended as part of the patch for just Application Control 64-bit agent.

The following command installs the base version of the Application Control Agent (MSI) and the Application Control patch file (MSP) simultaneously:

```
msiexec.exe /i ApplicationManagerAgent64.msi  
PATCH=c:\fullpath\ApplicationManagerAgent64.msp
```



A base version must be installed before the patch file can be applied.

If the patch file contains driver or hook files that are currently in use on the machine the patch is being applied to, you are informed that a reboot is required. If you chose to continue, the system is restarted when the patch has been applied.

For further information about installing and upgrading service packs using Management Center 8 FR4, see the *Management Center Install and Upgrade Guide*.

Installation Order and Dependencies

It is recommended that all components of a service pack are installed.

Rolling Back Service Packs

There are two ways to roll back, or uninstall Service Packs:

- Using the Windows Control Panel
- Using Management Center 8 FR4

If a service pack is uninstalled the installation reverts to the previous latest build, whether a service pack or base version.

Roll Back Service Packs Using Windows Control Panel

The procedure used to roll back service packs varies depending on the Operating System:

For Windows 7

Navigate to **Control Panel > Programs > Programs and Features > Installed Updates**. Highlight the selected patch and click **Uninstall**.

Roll Back Service Packs Using Management Center 8 FR4

1. In the Management Center console, select **Overview > Deployment Groups tab > Deployment Groups**.
2. Highlight the Deployment Group and select **Settings > Assigned Packages**.

The Assigned Packages work area displays a list of all the products and their associated packages.
3. Highlight the required Application Control service pack and click **Unassign** from the Actions menu.
4. Click **Review and Submit**.

The Submit Changes dialog displays.
5. Check the details are correct and click **Submit**.

The patch is unassigned based on the deployment group Installation Schedule.

Configuration

Application Control configuration files (AAMP) contain the rule settings for securing your system. The configuration files are installed on managed devices and serve as a policy checklist for the Application Control agent to assess how to handle file execution requests. When a file is executed, Application Control intercepts the request and performs a check with the configuration to find the appropriate matching rule and the required action to take. Other default policies specified in a configuration are also applied, for example, event filtering or handling for specific file extension types as well as general policies such as default rules, auditing rules, how message notifications are displayed, and archiving options.

Configurations are stored locally in different locations depending on your operating system and are protected by NTFS security: Windows 7 and above: C:\ProgramData\AppSense\Application Manager\Configuration.

In Standalone mode, configuration changes are written directly to the local AAMP file from the Application Control console. In Enterprise mode, configurations can be created and stored centrally in the Management Center database, and distributed to endpoints in MSI format via the Management Server. Configurations can also be exported and imported to and from MSI file format, which is useful for creating templates or distributing configurations using third-party deployment systems.

After creating or modifying a configuration, you must save the configuration with the latest settings to ensure that they are implemented.

Configuration Elements

Libraries

Application manager Library node allows you to create groups of items that can be used in configuration rules. Use the library to create a group of similar items to manage. Once your libraries have been created they can be assigned to rules and used to govern a group of users. Library nodes provide the following:

- **Group Management** - The Group Management node allows you to group a number of items such as Files, Folders, Drives, Signature Files, Windows Store Apps, and Network Connections for one particular application. You can then add this group to the Allowed and Denied Items lists in a rule.
- **User Privilege Policies** - The User node allows you to add User Privilege Policies to selectively promote or demote administrative rights for individual applications.

Rules

Rule nodes provide default settings for handling file executions and specific settings that apply to particular users, groups, or devices. Group, User, Device, Custom, Scripted, and Process Rules allow you to specify Security Level settings that specify restrictions that apply to users, groups, or devices matching the rule. Custom rules target combinations of particular users or groups operating on specific collections of devices. Scripted rules allow administrators to apply Allowed Items and Denied Items to users based on the outcome of a Windows PowerShell or VBScript script. Scripts can be run for each individual user session or run once per computer. Process rules allow you to manage access for the application to run child processes that might otherwise be managed differently in other rules. You can add Allowed Items, Denied Items, Trusted Vendors, User Privileges, and Browser Control to a rule.

- **Allowed/ Denied Items** — A sub-node list in each rule that you can populate and maintain with specific files, folders, drives, and digital signatures to provide an additional level of granularity for controlling file execution requests. For example, items that Trusted Ownership checking normally denies can be allowed for the users or devices targeted in the rule. Likewise, files that would normally be allowed can be denied.
- **Trusted Vendors** — A sub-node list in each rule that you can populate with digital certificates issued by trusted sources. Files that fail Trusted Ownership checking are checked for the presence of digital certificates and allowed to run when a match is made with the Trusted Vendors list. For example, a highly restricted user might be prohibited under normal rule conditions from introducing executable files on the system, but may be required to download and run software updates from a particular source from time to time. If the downloaded file includes a digital certificate that matches a certificate in the Trusted Vendors list, the file is allowed to run.
- **User Privileges** - A sub-node list in each rule that you can populate with applications, components, and web installations for you to apply User Privilege Policies to. User Privilege Policies allow you to selectively promote or demote administrative rights for individual applications, components, and web installations.
- **Browser Control** - A sub-node list in each rule that you can populate with URLs to which you can apply URL redirection. You can also specify URLs that open an elevated instance of Internet Explorer, and allow the elevation to administrative privileges for ActiveX installers from particular domains.

Default Configurations

Application Control is ready to manage your security as soon as you install the agent and a configuration on client computers. A default configuration loads when you run the console and can be used for immediate protection on all client computers to which the configuration is deployed. This configuration blocks any file with an untrusted owner and prevents non-administrative users accessing executables on non-secure locations, including network locations and removable media.

The default configuration can be saved directly in Standalone mode to the client computer via the console or saved to the database of the deployment mechanism when operating in Enterprise mode ready for deployment.

Protection

- All application and process execution requests are checked against the Application Control rules before access is granted.
- All application and process network access requests are prohibited unless allowed by Application Control rules.
- Members of the Local Administrators group are granted unrestricted access to applications.
- Members of non-administrative user groups are granted restricted access to applications.
- CMD.exe is blocked except when run by batch files.
- MSI, WSH and Registry Files are validated against the Application Control rules.
- Windows Installer (msiexec.exe) is allowed to run all child processes with the DLL and EXE extensions.

Default Configuration Settings

Setting		Value	Description
Advanced Settings	Policy Settings	General Features <ul style="list-style-type: none"> • Make local drives allowed by default • Ignore restrictions at logon • Allow cmd.exe for batch files • Allow self-extracting ZIP files • Ignore restrictions during Active Setup • Prohibit files on removable media 	<ul style="list-style-type: none"> • Ignore restrictions at logon delays the implementation of the Application Control rules until logon is complete to avoid disrupting or preventing the logon process. This option allows logon scripts to run. • While cmd.exe and self-extracting ZIP files are usually blocked as potential loopholes for attempts to breach security, this option allows CMD and ZIP files to run for legitimate files Application Control rules.
		Validation <ul style="list-style-type: none"> • Validate MSI (Windows Installer) Packages • Validate WSH (Windows Script Hosts) • Validate registry files 	System process validation can affect performance and is disabled by default.

Setting		Value	Description
			<ul style="list-style-type: none"> Application Control validates MSIs, Registry files, and WSH files against the rules by default. Otherwise, they are ignored unless they are specified in the rules themselves. Turn these options off only if you trust these types of files running or you have adequate protections in place in the Application Control rules or by some other method.
		Functionality <ul style="list-style-type: none"> Enable Application Access Control Enable Application Network Access Control Enable Privilege Management 	<ul style="list-style-type: none"> All Application Control functionality is enabled by default but you can disable any of these as part of any troubleshooting process.

Setting		Value	Description
			<ul style="list-style-type: none"> We recommend disabling any functionality which you do not want to use.
	Application Termination	Settings for closing and terminating applications. Set triggers, warning message behavior to users, and warning message notifications.	Disabled by default.
Libraries	Group Management Node	For creating reusable groups of applications for assigning to Rules.	No default settings.
	User Privilege Policies	Reusable User Privilege Policies that elevate or restrict user privileges. For assigning to files, folders, signatures, drives and application groups in Rules.	No default settings.
	Administrator	Local Administrator Group rule for managing access to applications for local administrators.	<ul style="list-style-type: none"> Security level set to Unrestricted. No other default settings are applied.
	Everyone	Group rule for all system users unless a user matches other rules with higher priority settings.	<ul style="list-style-type: none"> Security level set to Restricted. AppSense Program Files directories are added to Allowed Items. No other default settings are applied.

Setting		Value	Description
	Process	Windows Installer (msiexec.exe) <ul style="list-style-type: none"> • *.EXE • *.DLL 	<ul style="list-style-type: none"> • All EXE and DLL files are allowed to run when spawned by msiexec. • This rule does not manage access to msiexec. You must manage access to msiexec in another rule.

Maintain Configurations

Create Configurations

To create a new configuration, click **File > New**.

A new configuration displays and automatically provides the following protection by default:

- Applications not stored on local hard drives are prohibited. For example, applications on network drives and removable media are prohibited. Applications that are not owned by the administrator are prohibited. For example, any applications copied onto the computer's hard drives by a non-administrator are prohibited.
- All administrators can run any applications.

You must save a new configuration before the default settings are implemented.

Import Configurations

Configurations can be imported in to Application Control.

1. Click **File > Import & Export > Import Configuration from MSI**.
The Open dialog displays.
2. Navigate to the location of the MSI, select it and click **Open**.

The configuration opens in the Application Control console.

Export Configurations

Configurations can be exported from Application Control.

1. Click the **File > Import & Export > Export Configuration as MSI**.
The Save As dialog is displayed.
2. Navigate to the location to where you want to save the MSI and click **Save**.

Save Configurations

The following options for saving configurations are available from the File menu.

Save

- **Save and continue editing** - Saves the configuration and keeps it locked whilst open for editing. Any changes that have been made are not committed to the configuration and it cannot be deployed while locked.
- **Save and unlock** - Save the configuration and unlock it ready for deployment.
- **Unlock without saving** - Unlocks the configuration without saving any changes.

Save As

- **Live configuration on this computer** - Replace/update the configuration on the local computer with the currently open configuration.
- **Configuration in the Management Center** - Save the configuration in the package store on the selected Management Server.
- **Configuration in System Center Configuration Manager** - Saves your configuration to the specified System Center Configuration Manager server.
- **Configuration in Group Policy** - Allows you to create the configuration in a selected Group Policy store.
- **Configuration file on disk** - Save the configuration to disk.

Test Configurations

Set up a test user set up before proceeding with this task. The test account must not be one of the Trusted Owners in the configuration.

1. Log on, as the administrator, to an endpoint with the relevant Application Control configuration installed.
2. Start Application Control.
3. In the navigation tree, navigate to **Rules > User**.
4. Click Add Rule on the Rules ribbon and select User Rule.

The Add User Rule dialog displays.

5. Click **Browse**.

The Active Directory Select Users dialog displays.

6. Click **Advanced**.
7. Click **Find Now**.

The search results display in the bottom part of the dialog.

8. Scroll down to locate the test user, select and click **OK**.

The Select Users dialog displays with the test user displayed in the object name.

9. Click **OK**.

The User rule work area displays the newly created test user.

10. Save the configuration.
11. Log off as the administrator.
12. Log on as the test user to see Application Control working.

Group Policy Configurations

Group Policy provides centralized management and configuration of operating systems, applications, and users' settings in an Active Directory environment. Application Control uses Group Policy functionality to save and deploy configurations to any machine in a specified organizational unit (OU) in a domain without the need for additional infrastructure. To use Group Policy you must first install the Remote Server Administration Tools.

For more information see [Installing or Removing the Remote Server Administration Tools Pack](#).

To add an Application Control configuration file to a GPO, you must first add a Domain to the selectable list accessed from the Select Domain dialog. For more information, see Adding Selectable Domains to Your List.

If required, you can use the following command to install the Group Policy Management Console using PowerShell:

```
Import-Module ServerManager (2008 Server and above)
Add-WindowsFeature -Name GPMC
```

Add Selectable Domains to Your List

Add a domain to your list of selectable domains using the Select Domain dialog. Once the domain has been added you can then apply the Configuration to a GPO (Group Policy Object) on that domain.

1. From the File menu, choose **Save As** or **Open** and select **Configuration in Group Policy**.

The Select Domain dialog displays.

2. Select the Add icon from the toolbar.

The Add Domain dialog displays.

3. Enter the name of the domain to be added to the list. You must have the appropriate rights on the domain that you are adding.
4. Click the **Add** button.

The domain is added to your list and is ready to be selected.

Deploy Configurations Using Group Policy Objects

When a configuration is complete and deployed, the Client Side Extension copies the configuration into the Application Control %ProgramData% structure together with a merge_manifest.xml file. The Application Control Agent is notified of the update and the merge_manifest.xml file copied into the merge folder so merging can occur. The configuration is then applied to your endpoints.

Once the configuration is saved to the Group Policy Object (GPO), the deployment of that configuration is dependent on your organization's Group Policy settings.

Application Control supports the merging of multiple configurations deployed using Group policy. Each GPO may hold only one configuration; for multiple configurations to be deployed you need the same number of GPOs. If all GPOs reside in the same level in Active Directory, link order affects how configurations are merged, with the lowest number being the Base Configuration.



By default, computer Group Policy is updated in the background every 90 minutes, with a random offset of 0 to 30 minutes.

Save Configurations to a GPO

To save configurations to a Group Policy object, you must have an account that has read and write permissions in the area within the Active Directory (AD) you are working in. You can only save to that area and the policy applies only to the computers in it.

A configuration must be created with the Application Control console and a GPO must have been created within an Organizational Unit (OU) in the selected domain.

1. Create your AppSense Application Control Configuration file (AAMP).
2. Select **File > Save As > Configuration in Group Policy**.

The Select Domain dialog displays.

3. Highlight your selected Domain and click Connect.

If the domain you are saving to is not available from the list, the domain needs to be added.

See [Add Selectable Domains to Your List](#).

4. Navigate to your OU. You must have the appropriate rights on the OU you select.
5. Select the GPO and click **Save**.
6. If a GPO does not exist, right-click on the target OU and select Create a GPO in this domain, and link it here.



On some endpoints, you can experience a delay when saving the GPO to your Active Directory (AD). This is because AD replication is required to run across multiple Domain Controllers and Application Control will be unable to find the GPO until replication has been completed.

The GPO containing the configuration is stored in the following location and can be identified by its unique GUID.

\\<Domain Controller>\SysVol\<domain.fqdn>\Policies\<guid for GPO>\Machine\AppSense

If more than one configuration is deployed to an endpoint using Group Policy, Endpoint Configuration Merging occurs and the merged_configuration.aamp takes precedence over any existing configuration. For further information.

See [Endpoint Configuration Merging](#).

Global Settings

Use the Global Settings ribbon to define which defaults are to be applied to an Application Control controlled endpoint. These defaults form part of the Application Control configuration file (AAMP). Global Settings options include the following:

- [Trusted Owners](#)
- [Extension Filtering](#)
- [Application Termination](#)
- [Message Settings](#)
- [Archiving](#)
- [Policy Change Request](#)
- [Help Desk Portal](#)

Trusted Owners

During the rule matching process, Trusted Ownership checking is performed on files and folders to ensure that ownership of the items is matched with the list of trusted owners specified in the default rule configuration.

For example, if a match is made between the file you want to run and an allowed item, an additional security check ensures that the file ownership is also matched with the Trusted Owners list. If a genuine file has been tampered with or a file that is a security threat has been renamed to resemble an allowed file, trusted ownership checking identifies the irregularity and prevents the file execution.

Network folders/shares are denied by default. So, if the file resides on a network folder, the file or folder must be added to the rule as an allowed item. Otherwise, even if the file passes Trusted Ownership checking, the rule will not allow access.

Trusted ownership checking is not necessary for items with digital signatures because these cannot be imitated.

The list of Trusted Owners is maintained in the Trusted Owners dialog available from the Global Settings ribbon. Application Control trusts the following by default:

- SYSTEM
- BUILTIN\Administrators
- %ComputerName%\Administrator
- NT Service\TrustedInstaller

This means that, by default, Application Control trusts files owned by the BUILTIN\Administrators group and the local administrator. Application Control does not do group lookups for Trusted Owners – users who are members of the BUILTIN\Administrators are NOT trusted by default. Other users, even if they are members of the Administrators group, must be explicitly added to become Trusted Owners. You can extend the list above to include other users or groups.



When using Application Control for the first time, we recommend you use the default settings. To avoid complex customizations do not extend the Trusted Owners list or change any default settings.

The dialog contains the following options:

- **File Overwrite and Rename** - When the option **Change a file's ownership when it is overwritten or renamed** is selected, Application Control selectively changes the NTFS file ownership of executable files when they are overwritten or renamed.

Attempts by a user who is not a Trusted Owner to overwrite a file that is allowed due to Trusted Ownership or an Allowed Item rule, could constitute a security threat if the file contents have changed. Application Control changes the ownership of an overwritten file to the user performing the action, making the file untrusted and ensuring that the system is secure.

Likewise, attempts to rename a denied file to the name of an allowed item could also constitute a security threat. Application Control also changes the ownership of these files to the user who performs the rename action and ensures the file remains untrusted.

Overwrite and rename actions are both audited.

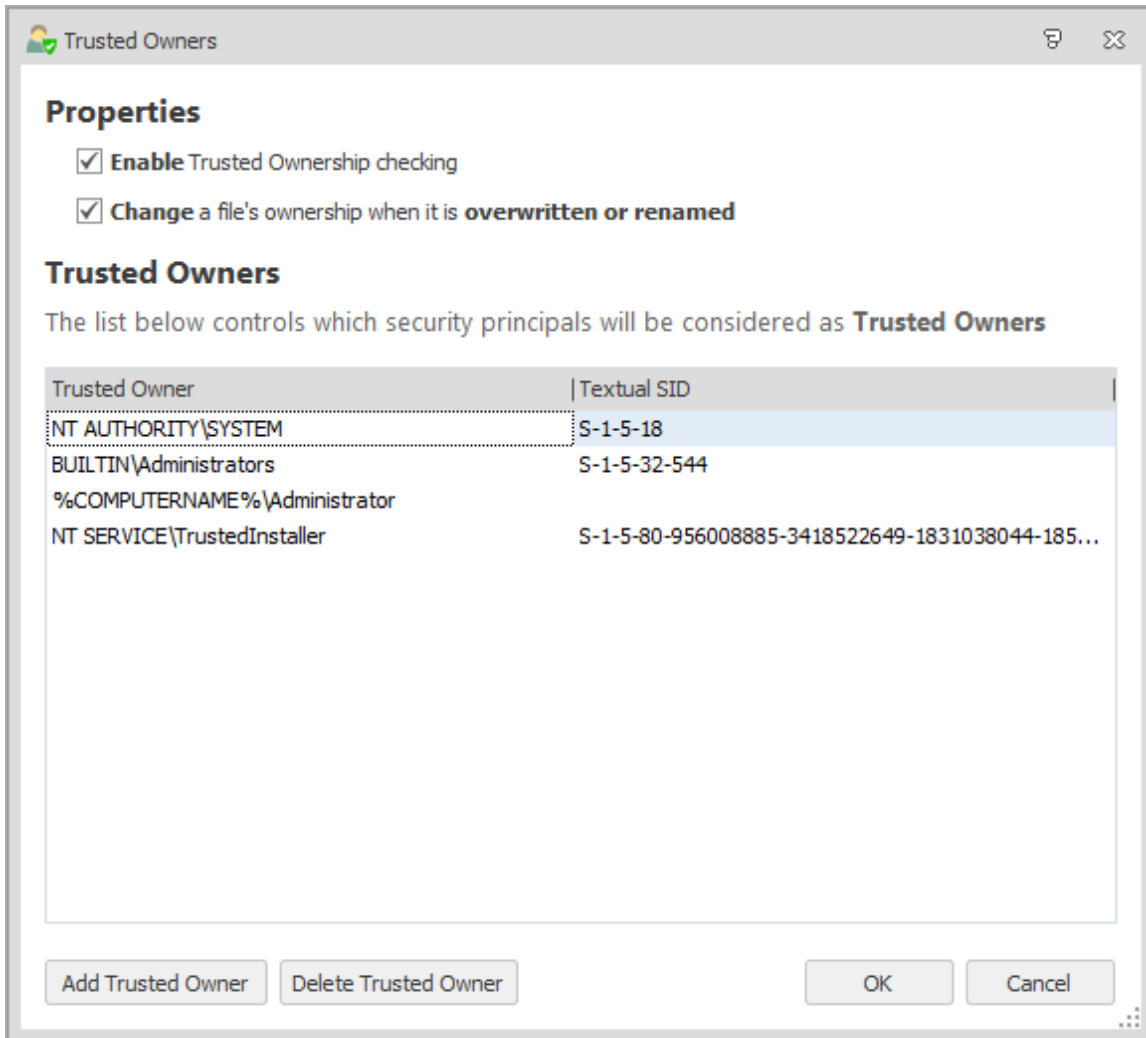
- **File Overwrite and Rename** - To ignore Trusted Ownership for individual files do one of the following:
 - Clear the Trust Ownership check box in the Allowed Items sub-nodes.
 - Assign self-authorization status to users and devices to allow the user to decide whether or not to allow a file to run.
 - Set the Self-Authorizing security level for a rule in the Group, User, Device, Custom, Scripted, and Process rule nodes.
 - Trusted Applications override restrictions resulting from matches with Denied Items.
 - Trusted Vendors override restrictions resulting from Trusted Ownership checking.

Whitelists

If you prefer to use a white list approach where nothing is allowed to run by default, clear the **Make local drives allowed by default** check box in the Policy Settings dialog available from Advanced Settings in the Global Settings ribbon. To make items allowed, add them to the Allowed Items folder of a configuration node.

If you use a Whitelist approach, ensure that you allow important system files to run by adding a Group rule for the Everyone group in which all of the relevant files or folders have been added to Allowed Items. Otherwise, many crucial executable files and DLLs, such as those that are stored in the system32 directory can be prevented from running and adversely affect correct system functioning.

Enable Trusted Ownership



To enable this feature, select **Trusted Owners** from the Global Settings ribbon and configure the required settings:

- **Enable Trusted Ownership checking** - Select to switch on Trusted Ownership checking. Selected by default.
- **Change a file's ownership when it is overwritten or renamed** - Select to change the ownership of any trusted allowed file which is overwritten by an untrusted user, who is not in the Trusted Owners list.



When a denied file is renamed by an untrusted user, in an attempt to bypass a denied item rule, the ownership is changed to the untrusted user. Once the ownership has changed, Trusted Ownership checking then prevents the file from being executed.

- **Trusted Owner** - The Trusted Owner details.

- **Textual SID** - The Textual Security Identifier of the Trusted Owner. For example, *S-1-5-32-544*.
- **Add Trusted Owner** button - Launches the Add Trusted Owners dialog. Enter or browse to select an account to add to the Trusted Owner list.
- **Delete Trusted Owner** button - Deletes the selected Trusted Owner.

Test Trusted Ownership

1. Introduce one or more applications using a test user account.
2. Copy one or more applications to the user's home drive or another suitable location, such as calc.exe from the System32 folder or copy a file from a CD.
3. Attempt to run a copied file. The application is denied because the files are owned by the test user and not a member of the Trusted Owners list.

You can verify the ownership of a file by viewing the Properties using Windows Explorer.

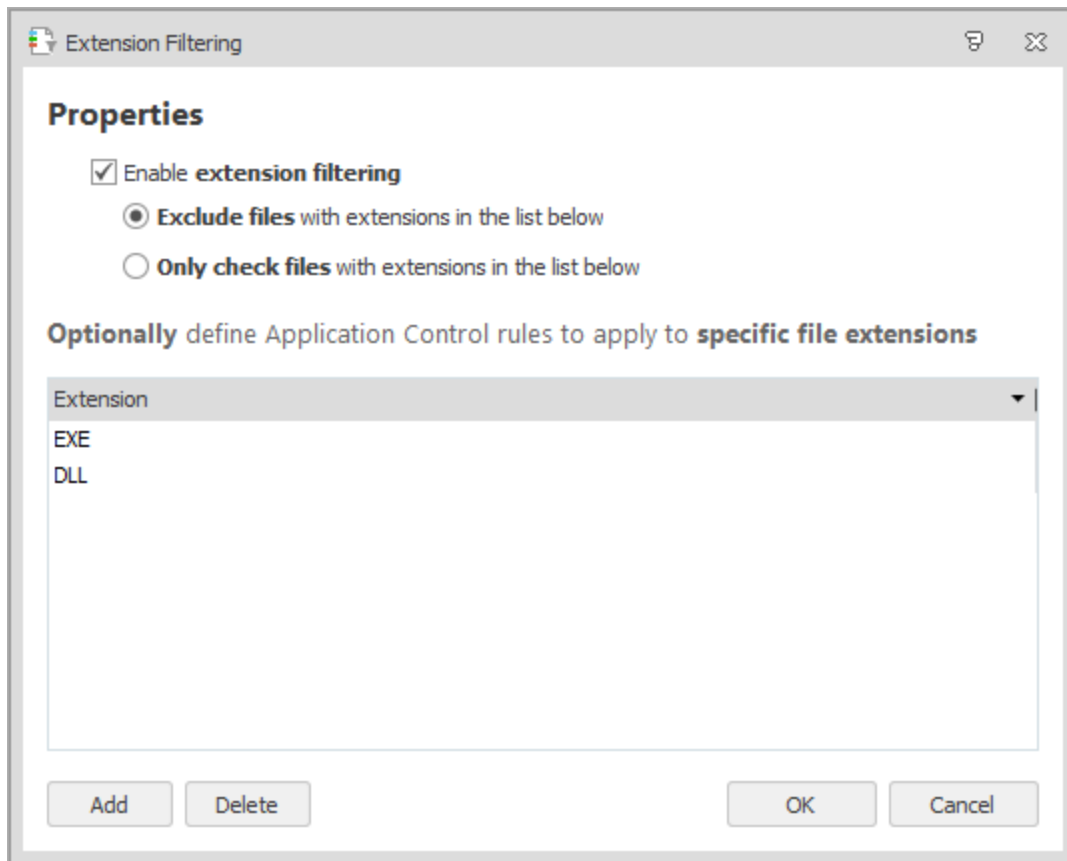
Extension Filtering

The Extension Filtering feature is used to determine whether the configuration should check certain file types or if it should ignore certain file types. The default configuration in Application Control does not have any extension filtering configured and therefore all executable code - irrespective of file extension - is checked. This is the most secure option because nothing can get past the agent unless it has been expressly configured in the remainder of the rules.

Enable Extension Filtering in the Extension Filtering dialog, which you access via the Global Settings ribbon. The Extension Filtering dialog contains the following options:

- **Exclude files with extensions in the list below** - Select to ensure that Application Control rules do not apply to the file types listed in the Extensions list.
- **Only check files with extensions in the list below** - Select to ensure that Application Control rules apply only to the file types in the Extensions list. All other file types are allowed to execute normally.
- **Extensions** - A list of file extensions to filter. You can add to and delete from the list.

Use the **Add** button to add the file extensions. Once the configuration is saved, the Application Control agent only checks the files with the specified extensions against the rules when execution requests occur against the computer that the configuration is deployed to.



Application Termination

Application Termination allows you to control triggers, behavior, and warning messages for terminating applications on managed endpoints. You can terminate applications gracefully, allowing the user to save work before closing, or force a termination. You can edit notification messages for each type of trigger individually.

Triggers for terminating an application include the following:

- The agent starts
- A new configuration is applied
- The computer IP address changes
- The connecting device changes

When a trigger is activated, processes are evaluated against the rules to determine if an application requires terminating. Rules with Self-Authorizing and Audit Only security levels are not evaluated because Self-Authorizing rules allow user discretion over application control and Audit Only rules do not apply Application Control control.

You can configure warning and terminate messages, but must abide by the following:

- The message caption must not be left blank, be a single line, and can contain up to 100 characters.
- The message body must not be left blank, can contain zero or more line breaks, and can contain up to 10000 characters.
- A separate message box must be used for each trigger type.

Application terminations can be audited and are associated with audit event 9017.

For further information, see [Auditing](#).

Application Termination is disabled by default. Enable the feature using the **Enable Application Termination** option on the Application Termination dialog, which you access in the Global Settings ribbon.

Configure Application Termination

1. Select **Application Termination** on the Global Settings ribbon.
2. Select **Enable Application Termination**.

3. Select the triggers to use for application termination:
 - **Configuration Applied** - Select to terminate an application according to the configuration that is applied.
 - **Computer IP address changed** - Select to terminate an application when the IP address of the computer changes, for example, moving between secure and insecure environments.
 - **Connecting device changed** - Select to terminate an application when the connecting device has changed, for example, changing from a desktop to a laptop in the same session.
4. Select the **Options** tab to define which actions are taken when an Application is terminated:
 - **Display an initial warning message** - Displays an initial warning message to inform the user that the denied application will be closed and to save any work. The time to close can be specified using the **Wait for...** option. Use in conjunction with the **Close Application and Terminate Application** options. If this is not used in conjunction with these options, a message is displayed and the denied application does not close.
 - **Close the application** - Closes the application following the initial warning message, allowing the user time to save their work.
 - **Terminate the application** - Terminates the denied application without giving the user a warning message
 - **Wait for...** - Specifies the time period, in seconds, between actions, and also the time between closing and terminating. The maximum period is 120 seconds.
5. To change the warning or termination message, select either the **Configuration Applied Message**, **IP Address Changed Message**, or **Connecting Device Changed Message** tabs, depending on the specified triggers. To configure warning and termination messages, use the following fields:
 - **Caption** - The text to display for the title of the warning or terminate message
 - **Message body** - The text to display for the body of the message.
 - **Note**
Environment variables are supported for both the caption and message body.
 - **Width** - Specify the width of the Application Termination message dialogs. The width is measured in pixels and applies to all messages. The default value is 0.
 - **Height** - Specify the height of the Application Termination message dialogs. The height is measured in pixels and applies to all messages. The default value is 0.
6. Click **OK**.
7. Save the configuration.

Application Control also has the ability to terminate applications through the [Time Limits](#) feature.

Set Up Application Termination for an IP Address Change

Use Application Termination to terminate an application when the IP address has changed. For example, when the IP address is out of the company range of IPs.

Step 1 - Set up the Application Termination Options

1. Select **Application Termination** on the **Global Settings** ribbon.
The Application Termination dialog displays.
2. Select the **Enable Application Termination** option. This is turned off by default.
3. Select the **Computer IP address changed** option on the **Triggers** tab.
4. Select the **Options** tab.
5. Do one of the following:
 - Select **Display an initial warning message** and **Close application** options. This will display an initial warning message, allowing the user to save any work and then close the dialog.
 - Select the **Terminate application** options. This will terminate the application without any warning. You can display an initial warning if required.
 - Select all three options.
6. Select the **IP Address Changed Message** tab.
7. Change the message if required.
8. Click **OK**.

Step 2 - Set up Device Rule for Working in the Office

1. This step is to set up the IP address range that is allowed for the work office.
2. Select the **Rules** node in the navigation pane.
3. Select the **Add Rules** drop-down arrow on the **Rules** ribbon and then select **Device Rule**.
A new Device rule is created under the **Device** rule node.
4. Right-click the new node and select **Rename**.
5. Enter an intuitive name, for example, *In Office*.
6. Right-click within the work area and select **Add Client Device**.
7. The **Add a Client Device** dialog is displayed.
8. Enter the IP address range that is allowed and click **Add**.

Step 3 - Set up Device Rule for Out of the Office

1. This step is to set up the IP address range that is not allowed, for example, when using VPN

from another location.

2. Select the **Rules** node in the navigation pane.
3. Select the **Add Rules** drop-down arrow on the **Rules** ribbon and select **Device Rule**.

A new Device rule is created under the Device rule node.

4. Right-click the new node and select **Rename**.
5. Enter an intuitive name, for example, *Out of Office*.
6. Right-click within the work area and select **Add Client Device**.
7. The **Add a Client Device** dialog is displayed.

Do one of the following:

- Enter the IP address range that is not allowed.
 - Enter *.*.*.* to imply all other IP addresses.
8. Click **Add**.

Step 4 - Save the Configuration

Message Settings

Message Settings are used to define how message boxes are displayed to users and to specify the content of messages displayed when users attempt to launch applications in violation of a defined configuration.

Application Control message boxes can be customized to meet the requirements of an organization by specifying company colors, logos and fonts. More advanced styling can be achieved by using the Cascading Style Sheet (CSS), which is editable direct from the Message Style tab. Styling is applied to all the Application Control message boxes but the content of the messages can be amended individually.

Use the options in the Message Settings dialog available from the Global Settings ribbon to configure settings for messages issued to users. You can set up messages for situations where access is denied, application limits have been exceeded, and for self-authorization. Time limits for application behavior can be specified with warning and denied messages.

Message Box Variables

The message box caption and text may contain user and system-wide environment variables, and include the following environment variables. Environment variables are not expanded during testing.

Environment Variable	Description
%ExecutableName%	Expands to the name of the denied application.
%FullPathName%	Expands to the full path of the denied application.
%DirectoryName%	Expands to the directory where the denied application is located.
%NetworkLocation%	Expands to the resolved IP address of the given hostname.

Configure Message Box Elements

For each type of message, define the following:

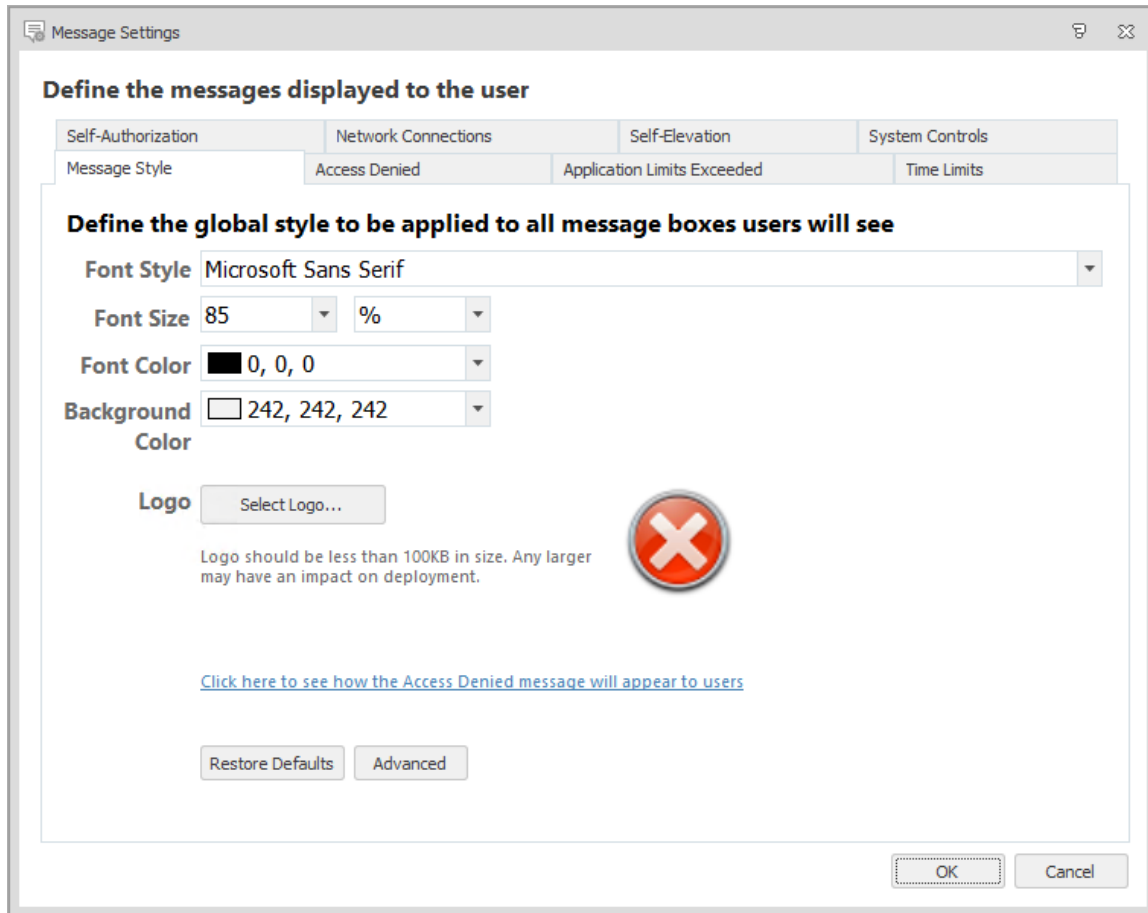
- **Caption** - The text to display at the top of the message. For example, you can change the default caption, Application Control, so that the user is not aware that Application Control has intervened.
- **Message body** - Enter the text to display in the body of the message.
- **Width** - Specify the width of the message dialog. The width is measured in pixels and applies to all messages. The default value is 0.
- **Height** - Specify the height of the message dialog. The height is measured in pixels and applies to all messages. The default value is 0.

When configuring messages, consider the following:

- Environment variables are supported for both the caption and the message. In addition to system environment variables it also supports the following for each file: %ExecutableName%, %DirectoryName% and %FullPathName%.
- When using hyperlinks in the message body, the full HREF attribute tag must be entered. For example, a
- If less-than or greater-than angle brackets are to be displayed in the message body, use **<** and **>** respectively.
JavaScript is not supported.

You also have the option to view how the message will appear to others. Select **Click here to see how the message will appear to users** - Displays the message with the caption and body specified.

Message Style



Application Control message boxes can be customized to meet the requirements of an organization by specifying company colors, logos and fonts. More advanced styling can be achieved by using the Cascading Style Sheet (CSS), which is editable direct from the Message Style tab. Styling is applied to all the Application Control message boxes but the content is managed for each message.

Define the required settings for all Application Control Message boxes:

- **Font Style** - Select the font type from the drop-down list.
- **Font Size** - Select the size of the font to be displayed. For specific font sizing, you can select the units by which the font is measured using the options available in the adjacent drop-down list.
- **Font Color** - Select the font color.
- **Background Color** - Select the background color of the message boxes.
- **Logo** - Use Select Logo to replace the default image on all Application Control message boxes. File sizes should be no larger than 100 kilobytes. Using logos may have an impact on the deployment of the configuration.
- **Restore Defaults** - Use Restore Defaults to undo any changes that have been applied to your message styles. For information on the options available, see Restore Defaults.

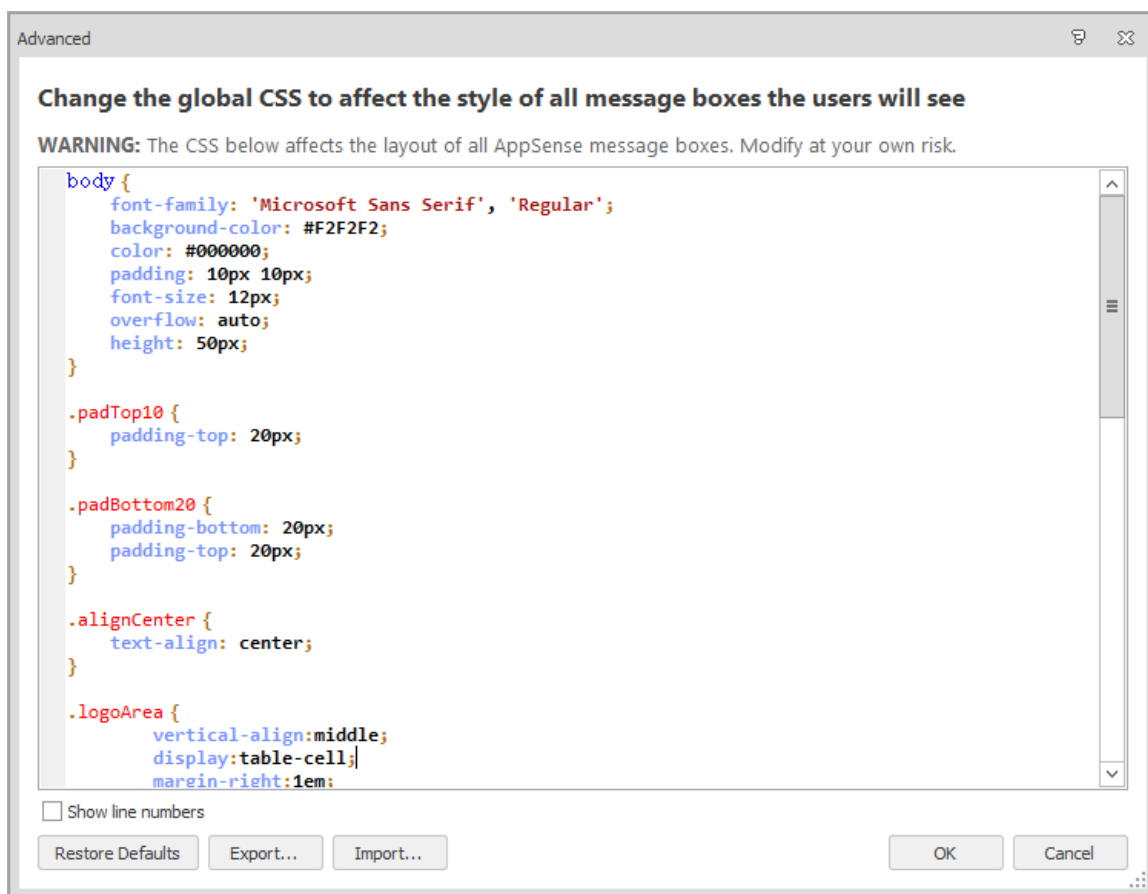
Use the **Click here to see how the message will appear to users** link to display an example of how the Access Denied message box will look when all the styles have been specified.

Restore Defaults

Use the Restore Defaults button to revert any changes that have been applied to your message styles. There are two options available:

- **Restore to AppSense defaults** - Select this option to restore the message box styling to the default settings. When this option is selected, the CSS and logo copied from the Application Control installed location overwrites any existing customization.
- **Restore to Configuration defaults** - Select this option to restore the message box styling to the styles specified in the configuration.

Advanced



Use the Advanced button to edit the message box style directly using CSS. When this option is selected the Advanced dialog displays. The dialog contains a basic CSS Editor, options to import, export and restore a CSS are also available.



It is recommended that an experienced user modifies the CSS. Any changes to styling will impact all Application Control message boxes. CSS3 is not supported.

Click the **Export** button and select a location to save the CSS file. When exported, the CSS file can be edited using another CSS editor and then re-imported when the amendments have been made.

Click the **Import** button and select the CSS file to open and use. The styles specified in the imported CSS will automatically overwrite any existing styles. These styles will take immediate effect but will not be applied until you save a configuration.

Self-Authorization

Self-Authorization is a security level within Application Control. Some applications require self-authorization by a user before they are allowed to run. You can specify the message displayed for both the initial message and the response. The self-authorization message displays when a self-authorizing user attempts to run a denied application and the file requires a user decision to run. The Response message displays when a self-authorizing user allows a DLL file that another application uses and the application may need to be restarted.

Configure the message that displays when self-authorization is required and the message that displays when an application has been authorized.

For more information, see [Security Level](#).

Access Denied

Message Settings

Define the messages displayed to the user

Self-Authorization	Network Connections	Self-Elevation	System Controls
Message Style	Access Denied	Application Limits Exceeded	Time Limits

Define the message shown when access is denied

Caption Application Manager

Message body %USERNAME% is not authorized to execute %ExecutableName%

Width 0 Height 0

[Click here to see how this message will appear to users](#)

Define the message shown when privileges are restricted

Caption Application Manager

Message body The current operation requires Administrative privileges.

Width 0 Height 0

[Click here to see how this message will appear to users](#)

OK Cancel

Access to applications can be denied or restricted for a user. Denied and restricted Items are specified in the Group, User, Device, Custom, Scripted, and Process rules.

Configure the messages that display when a user attempts to access an application that has been denied or when a user has insufficient privileges.

For more information, see [Rules](#).

Network Connections

The screenshot shows the 'Message Settings' dialog box with the 'Network Connections' tab selected. The dialog is titled 'Define the messages displayed to the user'. It contains several tabs: 'Message Style', 'Access Denied', 'Application Limits Exceeded', 'Time Limits', 'Self-Authorization', 'Network Connections', 'Self-Elevation', and 'System Controls'. The 'Network Connections' tab is active, showing options to define the action taken when a network connection is blocked. The 'Caption' field is set to 'Application Control - Application Network Access Control'. The 'Message body' field contains the text '%ExecutableName%' has been denied access to %NetworkLocation%. Below the message body, there are fields for 'Width' and 'Height', both set to 0. A link 'Click here to see how this message will appear to users' is provided. At the bottom right, there are 'OK' and 'Cancel' buttons.

Define the messages displayed to the user

Message Style | Access Denied | Application Limits Exceeded | Time Limits
Self-Authorization | **Network Connections** | Self-Elevation | System Controls

Define the action taken when a Network Connection is blocked

☒ Display a **message box** for blocked **network connections**

☐ Display a warning on **every** connection attempt

☒ Display a warning message **once**

☐ Wait for seconds between messages

Define the Network Connection Denied Message users will see

Caption Application Control - Application Network Access Control

Message body %ExecutableName% has been denied access to %NetworkLocation%.

Width Height

[Click here to see how this message will appear to users](#)

OK Cancel

The Network Connections message displays when a connection is blocked. Configure the following settings to determine the action taken when a network connection is blocked:

- **Display a warning message for blocked network connections** - Displays a message box for all blocked network connections. This option is enabled by default.
Selecting this option enables further settings and allows you to configure the content and dimensions of the connection denied message.
- **Display a warning on every connection attempt** - Displays a warning message every time a connection is attempted.
- **Display a warning message once** - Displays a message only on the first attempt per application within the same session.
- **Wait ... seconds between messages** - Specifies the number of seconds to wait before a new message is issued. Only one message displays per application within the specified period. No message displays for any subsequent attempts within the same period.

For more information, see [Application Network Access Control](#).

Application Limits Exceeded

The screenshot shows a 'Message Settings' window with a tabbed interface. The 'Application Limits Exceeded' tab is selected. The 'Caption' field contains 'Application Control'. The 'Message body' field contains '%USERNAME% has exceeded the application limit for %ExecutableName%'. Below the message body, there are 'Width' and 'Height' fields, both set to '0'. A link 'Click here to see how this message will appear to users' is present. At the bottom right are 'OK' and 'Cancel' buttons.

Define the messages displayed to the user			
Self-Authorization	Network Connections	Self-Elevation	System Controls
Message Style	Access Denied	Application Limits Exceeded	Time Limits

Define the message shown when the application limit is exceeded

Caption Application Control

Message body %USERNAME% has exceeded the application limit for %ExecutableName%

Width 0 Height 0

[Click here to see how this message will appear to users](#)

OK Cancel

The Application Limits Exceeded message displays when the user is denied access to an application that has reached an application limit.

Configure the content and dimensions of the message that is displays when application limits are exceeded.

For more information, see [Application Limits](#).

Self-Elevation

The screenshot shows the 'Message Settings' dialog box with the 'Self-Elevation' tab selected. The dialog is titled 'Define the messages displayed to the user'. It contains two sections: 'Define the Self-Authorization Message users will see' and 'Define the Response Message users will see'. Each section has a 'Caption' field, a 'Message body' text area, and 'Width' and 'Height' input fields. The 'Self-Authorization' section has a caption 'Application Control - Security Warning' and a message body '%ExecutableName% cannot run without your authorization. This action may be logged.' The 'Response' section has a caption 'Application Control - Authorized File' and a message body '%ExecutableName% is now authorized. Applications using this file may need to be restarted.' Both sections have a link 'Click here to see how this message will appear to users'. At the bottom right are 'OK' and 'Cancel' buttons.

Message Settings

Define the messages displayed to the user

Message Style | Access Denied | Application Limits Exceeded | Time Limits

Self-Authorization | Network Connections | **Self-Elevation** | System Controls

Define the Self-Authorization Message users will see

Caption Application Control - Security Warning

Message body %ExecutableName% cannot run without your authorization. This action may be logged.

Width 0 Height 0

[Click here to see how this message will appear to users](#)

Define the Response Message users will see

Caption Application Control - Authorized File

Message body %ExecutableName% is now authorized. Applications using this file may need to be restarted.

Width 0 Height 0

[Click here to see how this message will appear to users](#)

OK Cancel

Configure the content and dimensions of the message that displays when a user requests self-elevation.

The messages are displayed if the *Display a message box requiring a reason for Self-Elevation from the user* option is selected in the [Self-Elevation options](#).

1. In the Global Settings ribbon, select **Message Settings**.
2. Select the **Self-Elevation** tab.
3. In the Name field, enter the text to display for the self-elevation shortcut menu option.

The menu option is displayed when a user right-clicks a file with an extension on the [Self-Elevation file associations](#) list.

4. Configure the caption, content, and dimensions for the message that displays when a user requests self-elevation.
5. Click **OK**.

Time Limits

Message Settings

Define the messages displayed to the user

Self-Authorization | Network Connections | Self-Elevation | System Controls

Message Style | Access Denied | Application Limits Exceeded | Time Limits

Define action taken when the Time Limit is exceeded

☒ Display an **initial warning** message Wait for seconds between each action

☒ **Close** the application

☒ **Terminate** the application

Define the Warning Message users will see

Caption

Message body

Width Height

[Click here to see how this message will appear to users](#)

Define the Denied Message users will see

Caption

Message body

Width Height

[Click here to see how this message will appear to users](#)

OK Cancel

In Application Control, you can specify time limits for when applications can be accessed. For example, certain applications can be allowed to run only between 9 am and 5 pm, Monday to Friday. Two messages can be displayed:

- **Warning Message:** To inform the user if they are attempting to run the application outside of those hours.
- **Denied Message:** To inform the user if the time period has expired while the application is still running.

You can also specify whether the user is allowed to save their work before closing the application, or to just close the application upon the warning:

- **Display an initial warning message** - Select to display an initial warning message to the user when an application has exceeded time limits. Typically, this gives the user time to save their work and close the application. Use in conjunction with the Close application and Terminate application options. If you do not use this in conjunction with these options, only a message is displayed and application does not close.

- **Close the application** - Select to send a close message to the application. When most applications receive a close message they automatically give the user a chance to save their work. Select along with the **Display an initial warning** message option.
- **Terminate the application** - Terminate the application without allowing the user to save their work. Typically, this is used after the application has been sent a close message but has failed to terminate. Choose to select the **Display an initial warning message** or not, the application will terminate regardless.
- **Wait** - Specify the number of seconds to wait between each of the selected termination options. For example, if the user selects all three of the termination options and then selects 20 seconds, the warning message will be displayed, followed 20 seconds later by the close message and finally the application terminates after a further 20 seconds.

Configure the content and dimensions of the message that displays when time limits are exceeded.

System Controls

Message Settings

Define the messages displayed to the user

Message Style | Access Denied | Application Limits Exceeded | Time Limits

Self-Authorization | Network Connections | Self-Elevation | **System Controls**

Define the message shown when the uninstallation of a program is restricted

Caption Application Control

Message body The uninstallation of %ApplicationName% is not permitted.

Width 0 Height 0

[Click here to see how this message will appear to users](#)

Define the message shown when the clearing of an event log is restricted

Caption Application Control

Message body Clearing event log %EventLogName% is not permitted.

Width 0 Height 0

[Click here to see how this message will appear to users](#)

OK Cancel

System Controls are used to prevent users from:

- Stopping named services
- Clearing event logs
- Uninstalling or modifying specific applications

A message is displayed when the uninstallation of a program is restricted or when an event log cannot be cleared.

Configure the content and dimensions of the message that displays for both messages.

For more information, see [System Controls](#).

Archiving

Archiving is an optional function that allows you to copy any denied executables into a secure folder. When a user attempts to run an unauthorized executable, or an executable specified in the prohibited items list, Application Control can take a copy of each application that attempted to execute and place them in a secured file system or archive. This information can be used by an administrator to inspect the kinds of executable content that Application Control has blocked.

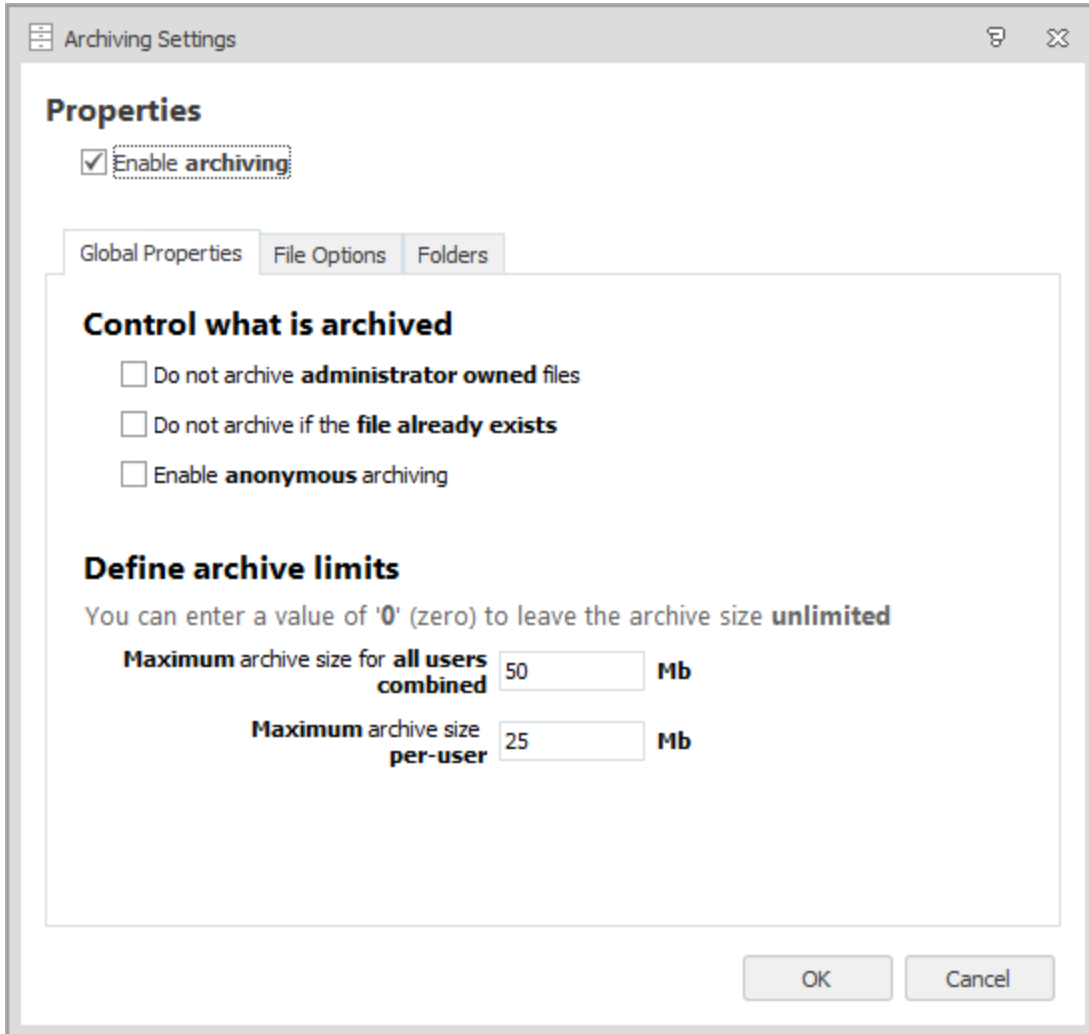
Blocked applications can often be files with false names such as winword.exe. The name alone does not tell the administrator much because these are typically other executables that have been simply renamed in an attempt by the user to get the application to run on the computer. Because Application Control takes a complete copy of each executable, the administrator can accurately assess each application and what impact they would have on the enterprise had they been allowed to run.



It is recommended that archived executables are checked in a secure environment to minimize the threat from viruses and malware.

Enable archiving by selecting **Enable Archiving** in the Archiving Settings dialog, which you access via the Global Settings ribbon.

Global Properties



The image shows the 'Archiving Settings' dialog box with the 'Global Properties' tab selected. The 'Enable archiving' checkbox is checked. Under 'Control what is archived', three options are listed: 'Do not archive administrator owned files', 'Do not archive if the file already exists', and 'Enable anonymous archiving', all of which are currently unchecked. Under 'Define archive limits', there are two input fields: 'Maximum archive size for all users combined' set to 50 Mb, and 'Maximum archive size per-user' set to 25 Mb. The dialog has 'OK' and 'Cancel' buttons at the bottom right.

Archiving Settings

Properties

☒ **Enable archiving**

Global Properties | File Options | Folders

Control what is archived

☐ Do not archive **administrator owned** files

☐ Do not archive if the **file already exists**

☐ Enable **anonymous** archiving

Define archive limits

You can enter a value of '0' (zero) to leave the archive size **unlimited**

Maximum archive size for all users combined 50 Mb

Maximum archive size per-user 25 Mb

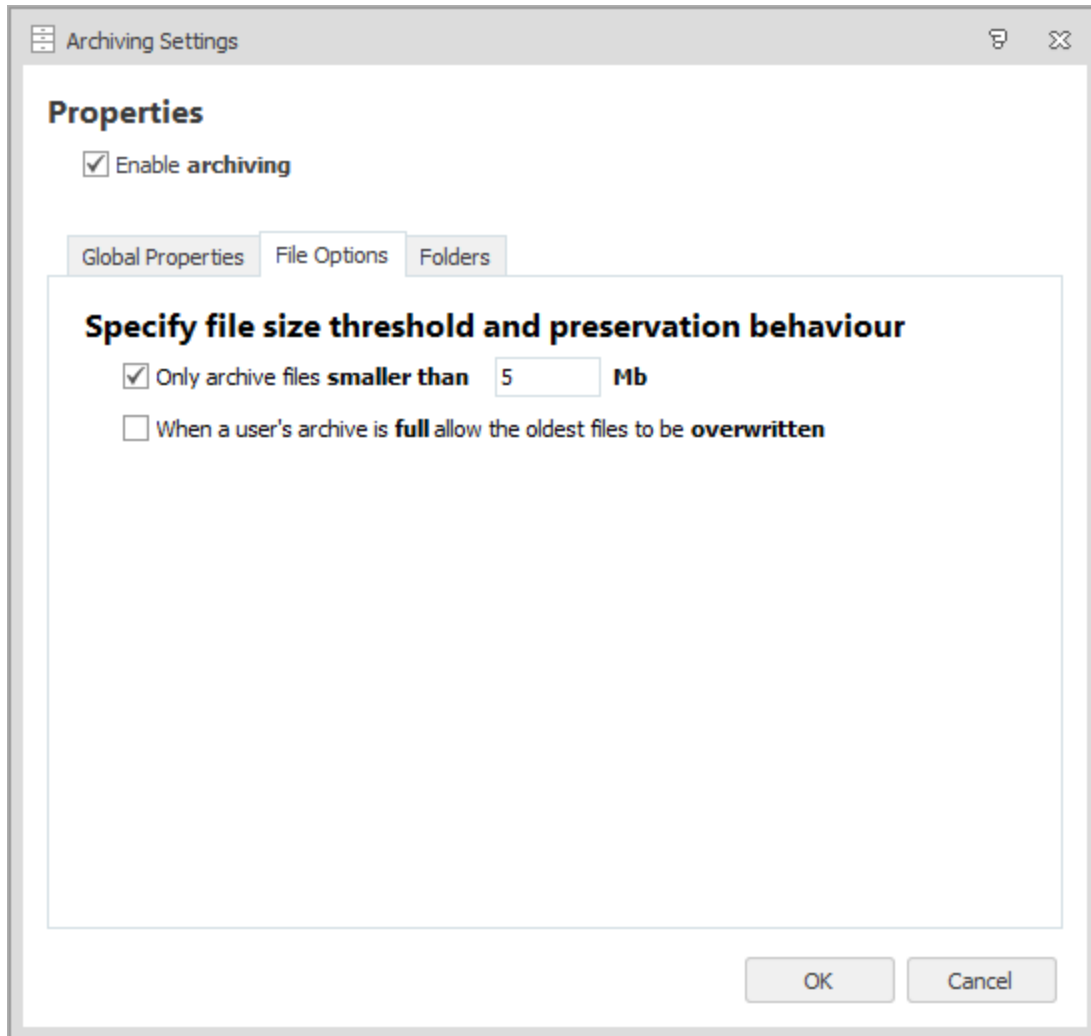
OK Cancel

Use the Global Properties tab in the Archiving Settings dialog to control what is archived and to define the maximum or minimum size of the archives, by selecting one or more of the following:

- Do not archive administrator owned files - Select to prevent Application Control from adding administrator-owned files to the archive. An example of a use case for this is when a user tries to execute regedit.exe and is blocked by the Application Control agent. It is unlikely you would require an archive of this file. However, it is useful to archive when the user attempts to execute their own copy of regedit.exe to determine what the application is and what effect it could have on the enterprise if it were to execute.
- Do not archive if the file already exists - Select to prevent Application Control from adding files to the archive that already exist in the archive, especially if the archive resides on the network. Duplicate entries are not created when this option is deselected. The existing archive entry is overwritten. This helps to save space, although it may result in inaccurate archiving as only one copy of an executable with the same name is ever retained.

- Enable anonymous archiving - Select to prevent Application Control from adding any user names to the archive. For example, if a user runs a downloaded file from the \$Home drive, the owner of the file is that user and also the archived filename contains the user's name as part of the path from which it was executed. If Anonymous archiving is selected, the owner of the file is changed to SYSTEM and any references to the user name are replaced with anonymous.
- Maximum archive size for all users combined - The maximum size in MB that combined users are allowed to reach before files are overwritten. A limit setting of zero (0) is interpreted as no limit.
- Maximum archive size per-user - The maximum size in MB that a single user archive is allowed to reach before files are overwritten. For example, if an archive path is specified as C:\archive\%username%, every user on the system has a separate archive under the C:\archive directory. It is this user archive that is subject to the user limit. The User Limit should not exceed the Total Limit. A limit setting of zero (0) is interpreted as no limit.

File Options

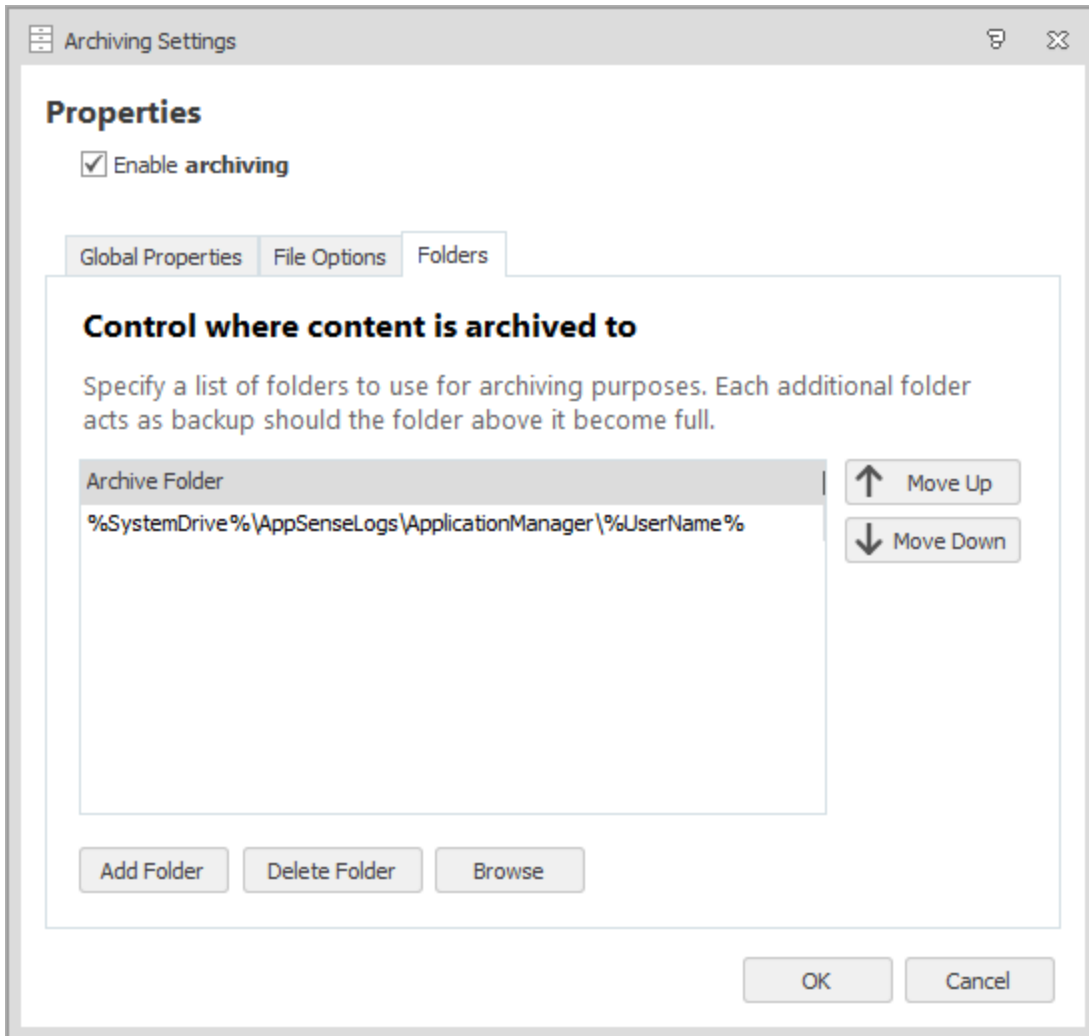


Use the File Options tab to specify file size thresholds and preservation behavior what is archived and to define the maximum or minimum size of the archives by selecting one or more of the following:

Only archive files smaller than - Limits the size of the files that are copied to the archive. This is particularly useful if a network archive is specified because copying large files to a network location is a potentially time consuming operation.

When a user's archive is full allow the oldest files to be overwritten - Select to allow Application Control to overwrite the oldest files in the archive in cases where the archive size has reached either the Total limit or the User limit. This is an easy way to ensure that the enterprise captures the most up-to-date information without using large amounts of data space for unauthorized applications.

Folders



Use the folders tab to specify a list of folders that can be used for archiving purposes, each of the folders can then be used to store backups.

The default location to place all archived files into is:

%SystemDrive%\AppSenseLogs\ApplicationManager\%UserName%

This places all archived files for a specific user in the same folder and the folder is named after the user making it easier to manage.

- **Archive Folder** - The list of folder paths to which archive files are copied. Archiving attempts to write to the first listed folder, if unsuccessful an attempt is made to archive to the next folder, if there is one in the list. This process continues until the folder list is empty or the archive action succeeds.

- **Move Up** - Moves the selected archive up the list of available archives. The order of the archive list is important as Application Control attempts to copy the file to the first archive in the list. If this copy fails, Application Control continues to make attempts to copy the file to the next archive location until it is successful.
- **Move Down** - Moves the selected archive down the list of available archives. The order of the archive list is important as Application Control attempts to copy the file to the first archive in the list. If this copy fails, Application Control continues to make attempts to copy the file to the next archive location until it is successful.
- **Add Folder** - Add an archive location to the list. The archive may contain environment variables. For example, %SYSTEMDRIVE%\Archive\%USERNAME% is expanded when Application Control attempts to archive the file. Each user has a personal archive.
- **Delete Folder** - Deletes the selected folder.
- **Browse** - Browse to the location where you want the archive to exist.

Policy Change Requests

Desktop and mobile users can use the Policy Change Request feature to request an update to an Application Control configuration via email or telephone. Endpoint users can make requests from a link on the Application Control Access Denied dialog or by manually using the Application Control Policy Change Request executable installed on their desktop.

Policy Change Request settings are configured per rule and are evaluated at session connect and when a configuration changes. The email address, telephone number, and text for change requests is set globally and used for all groups with the appropriate settings applied.

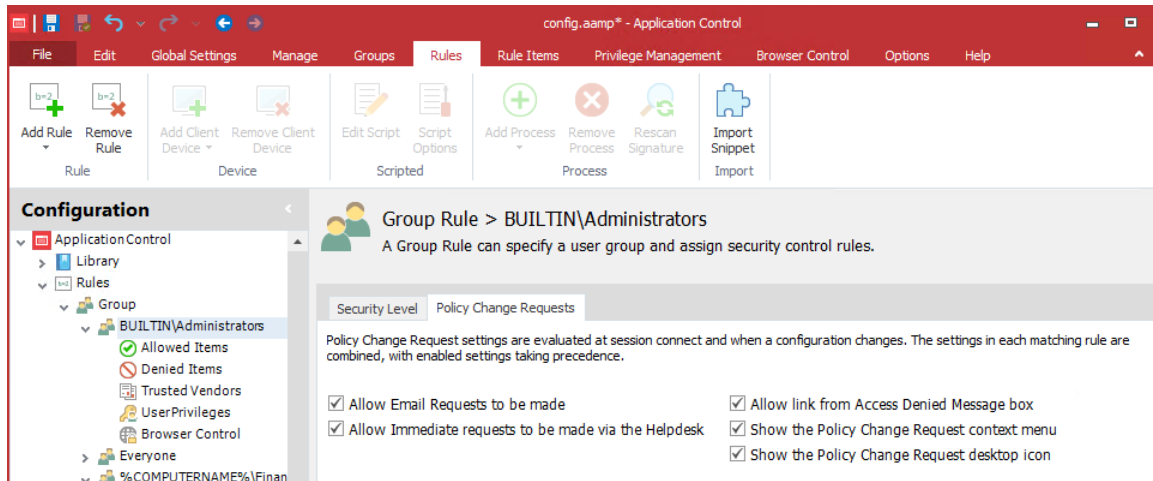


The Policy Change Request feature is only compatible with 32-bit and 64-bit versions of Internet Explorer 9, 10 and 11.

Upgrading Policy Change Request Settings

In 10.1 Policy Change Request behavior changed from being a global setting to being applied for each rule. This change prevents 10.1 agents processing change requests from endpoints with pre-10.1 configurations. To ensure the Policy Change Request feature continues to function correctly in 10.1, upgrade all configurations in the 10.1 Application Control console and redeploy.

Configure Change Requests for a Rule



Configure which request types and features are available to users for each rule. Policy Change Request settings are available for all rule types, apart from Process rules.

1. Select a rule in the navigation pane.
2. Select the **Policy Change Requests** tab.
3. Apply the required settings for the selected group:
 - Allow Email Requests to be made
 - Allow Immediate requests to be made via the Helpdesk
 - Allow link from AppSense Access Denied Message box
 - Show the Policy Change Request context menu
 - Show the Policy Change Request desktop icon

The detail for each setting is configured using the Policy Change Requests dialog, accessed from the Global Settings ribbon.

Configure Request Types and Methods

To configure request types and methods, select **Policy Change Requests** from the Global Settings ribbon.

Request Types

The screenshot shows the 'Policy Change Requests' dialog box with the 'Request Types' tab selected. The dialog has a title bar with a close button. Below the title bar, the text reads: 'Policy Change Requests. Allows users to request a policy change for access to, or additional user privileges for, **applications, installation files** and **Control Panel components**. Each item can be enabled on a per rule basis. The text visible to the user is set globally.' Below this text are two tabs: 'Request Types' (selected) and 'Request Methods'. Under the 'Request Types' tab, there is a section titled 'Define the settings used by the email and help desk request types. Request types are **enabled on a per rule basis**.' This section contains two sub-sections: 'Email requests:' and 'Immediate requests:'. Under 'Email requests:', there is a 'Mail To' field with the value 'itsupport@company.com'. Under 'Immediate requests:', there is a 'Help desk phone number' field with the value '+18005567789' and a 'Shared key' field with the value '*****'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

Configure email and immediate policy change requests in the **Request Types** tab on the Policy Change Requests dialog.

Email Requests

When a user is prompted to elevate their privileges to run an application, they can click a link in the Access Denied message box to request a permanent configuration change. When the user clicks the link, they are prompted to enter the reason for the change request, which is sent to the email address configured in the Application Control console.

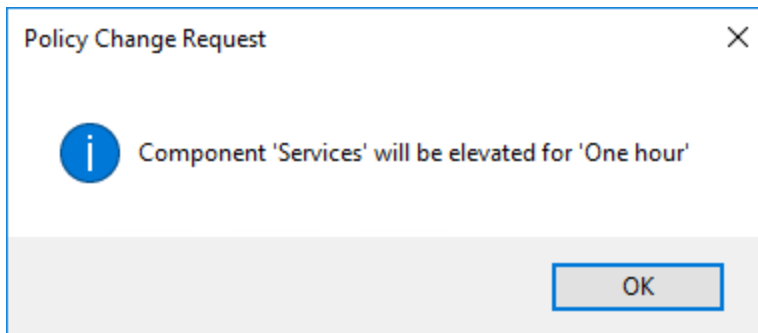
The Email Request function uses Messaging Application Programming Interface (MAPI) to send emails. An Application Control administrator reviews the request, and if the request is granted, updates the configuration and deploys the AAMP file.

To set up email change requests, enter the email address to which change requests are sent in the **Mail To** field.

Immediate Requests

Immediate requests allow users, typically mobile users, to request a permanent or temporary configuration change. When users click the immediate request link, they are provided with a phone number to call and issued with details of the request and a request code. The request code and the configuration change request are relayed to IT Support, who enter the details in the [Help Desk Portal](#). IT Support generate a response code and send it to the user to enter in the Policy Change Request dialog.

Users get three attempts to enter a response code. After three incorrect attempts the dialog closes and the changes are not applied. If configured, when the dialog closes, a 9091 event is raised. If the user requires further configuration changes, they must restart the process. If the code is entered correctly, users have elevated access to the application. Upon confirmation, users are presented with details of the elevation.



Configure the following fields in the Request Types tab:

- **Helpdesk Phone Number** - The number users are prompted to call to request the immediate configuration change.
- **Shared Key** - The shared key is an integral part in processing Immediate requests and is embedded in the configuration. The shared key must match in both the Application Control Console and the Help Desk Portal. If the shared keys do not match, a response code cannot be created and configuration change will not be authorized for deployment to the user's endpoint.

The shared key can be changed using the Help Desk Portal, however if the shared key is amended in the Portal, the same key must also be entered in the Application Control Console.

Once you have configured the Immediate Request settings in a configuration file, deploy it to your endpoints. Before the feature is fully activated, the Help Desk Administrator and Help Desk Operator roles must be assigned to members of your Support Team. Once you have deployed the configuration and assigned Help Desk Administrator role, the Help Desk Administrator can assign or remove additional Help Desk Operators and/or Administrators.

Request Methods

Policy Change Requests

Allows users to request a policy change for access to, or additional user privileges for, **applications, installation files** and **Control Panel components**. Each item can be enabled on a per rule basis. The text visible to the user is set globally.

Request Types Request Methods

Define the text that appears to the end user for each method of making a change request. Each of these options is **enabled on a per rule basis**.

Link from **Access Denied Message Box**

Link Text

[Click here to see how this message will appear to users.](#)

Policy Change Request **menu item**

Text

This item is included in the menu shown when a user right-clicks an item for which a policy change can be requested.

Policy Change Request **desktop icon**

Text

OK Cancel

In the Request Methods tab, configure the text for policy change request items:

- Message Box Link Text - The text for the request link in the Link from Access Denied Message Box. The default text is *Click here to request access to this application*.
- Menu Item Text - The text for the menu item, displayed when a user right-clicks an item that is eligible for policy change requests.
- Desktop Item Text - The name of the Policy Change Request desktop icon. Users can use the icon to open the Application Control Policy Change Request dialog and create change requests.

Help Desk Portal

The Help Desk Portal is a browser based interface that allows IT Support to process an immediate configuration change request. The immediate configuration change requests are as a result of your endpoint users attempting to use a file or process that is prevented by their existing Application Manger configuration. The Portal is accessed from the following URL and used to generate a response code that will allow the configuration change to occur once the response code is entered by the user.

```
http://<MachineName>/OnDemand
```

The machine name is that of the endpoint on which the service is deployed.

The Help Desk Portal consists of two tabs, access to which is determined by the roles you have assigned to members of your IT Support team:

- **Config Requests** - This tab is used to generate the response code when an endpoint user contacts IT Support to request an immediate change to a configuration. The Help Desk Operator asks the endpoint user for all the details required to fill in the fields relevant to their request together with their request code.

When the endpoint user is relaying the information relating to their change request to the Help Desk, they are to provide the details as presented on their screen. If the endpoint user's dialog has "--Not Available--" for the Manufacturer, the Help Desk operator must leave their corresponding Manufacturer field blank.

- **Administration** - Use this tab to add or remove roles assigned to IT Support. This tab also provides administrators with the facility to change the shared key.



Although the shared key can be changed any time a new configuration is created, it is recommended that you do not do this because it can adversely affect performance.

Help Desk Portal Roles

Two portal job roles can be assigned to users in IT Support: Help Desk Administrator and Help Desk Operator. When your selected IT Users logs in to the portal using their Windows or Domain credentials, they are automatically redirected to the page on the Help Desk Portal associated with their assigned role.

Help Desk Administrators (HDAs)

The Help Desk Administrator role grants the selected user the privileges to perform the following tasks within the portal:

- Add or remove Users or Groups to the list of Help Desk Administrators.
- Add or remove Users or Groups to the list of Help Desk Operators
- Upload a new shared key

Help Desk Operators (HDOs)

The Help Desk Operator role allows the selected user to grant a configuration change and provide an activation code, following the endpoint user successfully answering a number of questions as specified in the Help Desk Portal.

By default, there are no users with the Help Desk Operators role automatically selected. Select the users by logging into the Portal as a Help Desk Administrator and assign the Roles.

Help Desk Portal Configuration

Use the Administration tab on the Portal to configure the shared key, add additional Help Desk Administrators and Help Desk Operators. As the Administrator, you are automatically assigned the Help Desk Administrator role as you are in the Local Administrator Group or a Domain Administrator. Both the Config Requests tab and the Administration tab are available.

When the roles are assigned and the selected users access the Help Desk Portal, they will be prompted to enter their Windows Login credentials. The credentials contain details of the assigned role and open the Portal according to the privilege level granted to the user.

Users with membership in the BUILTIN\Administrators group, have Administrator access to the Help Desk Portal. This access is implicit and cannot be changed, therefore the BUILTIN\Administrators group does not appear in the list of users or groups with access.

Amending the Shared Key

As a Help Desk Administrator, you can amend the existing shared key using the Shared Key field.

The shared key must match in both the Application Control Console and the Help Desk Portal. If the shared keys do not match, a response code cannot be created and configuration changes will not be authorized for deployment to the user's endpoint.

Any change made to the Shared key is logged under audit number 9040.

Assigning the Administrative User Role

Use the Add User link to find and select the additional administrative users or remove an existing administrator by highlighting the username in the list and clicking Delete. The user will not be removed from your organization's Active Directory but will instead be removed from the list of Help Desk Portal Administrators.

Assigning the Help Desk Operator Role

Select which users are to be granted the Help Desk Operators role. Use the Add User link to find and select your operators or remove them from your list by using the **Delete** button.

Help Desk Portal Request Logging

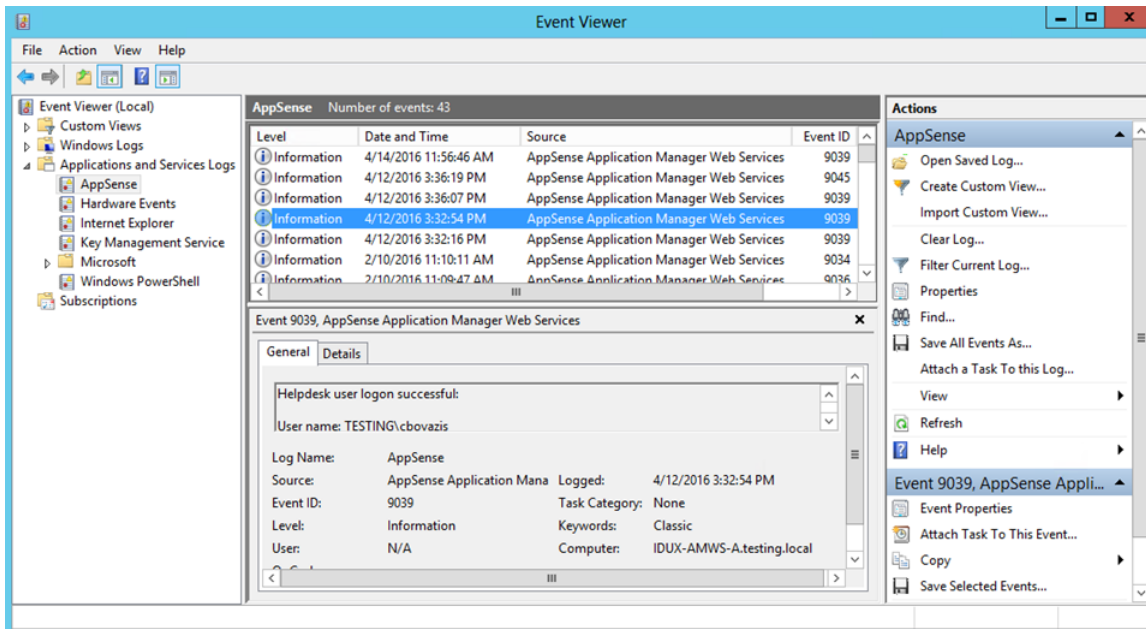
Application Control raises auditing events for Policy Change Request events performed on Application Manager Web Services, such as when a user logs on to the Help Desk Portal or when an administrator authorizes a Help Desk Operator. The following events are raised to the local event log of the server that is hosting Application Manager Web Services:

Event ID	Description
9039	Help Desk User logon is successful.
9040	Shared key has been modified.
9041	Help Desk response code has been generated.
9042	Authorized Help Desk User has been added.
9043	Authorized Help Desk User has been added.
9044	Authorized Help Desk User information has been updated.
9045	Help Desk User logoff is successful.
9046	Help Desk User logon failed
9047	Help Desk failed to get shared key.
9048	Help Desk failed to generate response code.
9049	Help Desk User authorization was unsuccessful.
9050	Help Desk failed to check local members of Administrators group.

View Help Desk Portal Request Events

1. On the server hosting Application Manager Web Services, open Event Viewer.
2. In Event Viewer, expand **Applications and Service Logs**.
3. Select the **AppSense log**.

Event Viewer lists the AppSense events in the center pane.



Manage

Use the Manage ribbon to manage which defaults are to be applied to all your Application Control configuration. You can also use this ribbon to specify how a deployed configuration is to be monitored. The Manage ribbon gives access to the following features:

- [Advanced Settings](#)
- [Signature Hashing](#)
- [Auditing](#)
- [Configuration Profiler](#)
- [Change Tracking](#)
- [Privilege Discovery Mode](#)
- [Privilege Discovery Results](#)

Advanced Settings

Advanced Settings are accessed from the Manage ribbon and allow you to assign global settings to the Application Control Configuration file. Specify the required global components using the Policy Settings and Custom Settings tabs.

Policy Settings

Archiving Settings

Properties

☒ **Enable archiving**

Global Properties | File Options | Folders

Control what is archived

☐ Do not archive **administrator owned** files

☐ Do not archive if the **file already exists**

☐ Enable **anonymous** archiving

Define archive limits

You can enter a value of '0' (zero) to leave the archive size **unlimited**

Maximum archive size for all users combined 50 Mb


Maximum archive size per-user 25 Mb

OK Cancel

Application Control Policy Settings are available in the Advanced Settings dialog and provide general Application Control settings to apply to all application and process execution requests.


General Features

Option	Description
Make local drives allowed by default	Select this option to make Application Control configurations blacklists. Everything on the local drive is allowed unless it specified in the Denied Items list, or it fails trusted ownership. Deselect this option to make the configuration a whitelist. Everything on the local drive is blocked unless it is specified in the Allowed Items list.

Option	Description
	A whitelist configuration is the most secure. However, this type of configuration is time consuming to make and can affect the client stability as all unspecified applications are blocked.
Allow cmd.exe for batch files	It is expected that administrators explicitly prohibit cmd.exe in their Application Managers configuration. When cmd.exe is denied and ' Allow cmd.exe for batch files ' is disabled, batch files will be evaluated and blocked if they fail the Application Managers policy. If the option is not selected and cmd.exe is explicitly denied, all batch files are blocked, they aren't even evaluated. If this option is selected and cmd is explicitly denied, cmd.exe still can't be run on its own, but batch files are evaluated against Application Control rules. If cmd.exe is not explicitly denied, all batch files run no matter whether this option is ticked or not.
Ignore restrictions during logon	During logon the computer may execute a number of essential applications. Blocking these can cause the computer to function incorrectly, or not at all. Hence, this option is selected by default.
Extract self-extracting ZIP files	<p>A self-extracting file is an executable that contains a ZIP file and a small program to extract it. These files are sometimes used as an alternative to installing an application by an MSI file. A number of administrators prefer applications to only be installed by an MSI file.</p> <hr/> <p> Only self-extracting EXEs formatted using the ZIP specification are supported. For additional information, see ZIP Specifications</p> <hr/> <p>The Extract self-extracting ZIP files option allows a denied executable file, which is a self-extracting ZIP file, to be extracted by the ZIP Extractor. If this option is deselected (the default setting) the file is subject to the normal rule processing as though it is an executable file. Once the contents have been extracted, any executable content it contains is still subject to the normal Trusted Ownership checks and is prevented from executing if the user is not a Trusted Owner. This is useful for scenarios where the self-extracting ZIP file may contain non-executable content such as a document that the user requires. By default, this option is deselected, and the self-extracting ZIP file is treated as a standard executable and can be prevented from executing (and hence extracting its contents) subject to the normal rule processing.</p>
Ignore Restrictions during Active Setup	By default, all applications which run during Active Setup are subject to Application Control rules. Select this option to make these applications exempt from rules checks during Active Setup phase.

Option	Description
Prohibit files on removable media	Deselect this option to remove the restrictions on removable media. Removable media is whatever the call to GetDriveType determines it to be. Due to the nature of removable media, the drive letter may change depending on how an endpoint is setup. For example: On one computer the removable media drive may be identified as the E: drive and on another F:

Validation

Option	Description
Validate System processes	Select this option to validate any files executed by the system user. Note that it is not recommended to select this option as it increases the amount of validation occurring on the endpoint computer and can block crucial applications from running. Selecting this option means all executables launched by the system are subject to rule validation.
Validate WSH (Windows Script Host) scripts	Selecting this option specifies that the command line contents of scripts ran using wscript or cscript are subject to rule validation. Scripts can introduce viruses and malicious code. It is recommended to validate WSH scripts.
Validate MSI (Windows Installer) packages	MSI files are the standard method of installing Windows applications. It is recommended that the user is not allowed to freely install MSI applications. Selecting this option means all MSIs are subject to rule validation. Deselecting this option means that only the Windows installer itself, msixec.exe, is validated by the Application Control rule processing, and not the MSI file that it is trying to run.
Validate Registry files	Select this option to enable rule validation for regedit.exe and regini.exe. Deselecting this option means that the regedit.exe and regini.exe, is no longer blocked by default. Additionally, the .reg script, the regedit.exe and regini.exe it is trying to run is no longer validated by Application Control rules processing.  It is not recommended to allow users to access the registry or registry files.
Validate PowerShell scripts	When enabled, this setting denies powershell.exe and powershell_ise.exe. However, if a PowerShell script (PS1 file) is found on the command line, then, it is subjected to a full rules check to see if it is configured for elevation, allowed, or denied.
Validate Java archives	When enabled, this setting denies java.exe and javaw.exe. However, if a Java archive (JAR file) is found on the command line, then, it is subjected to a full rules check to see if it is allowed or denied.

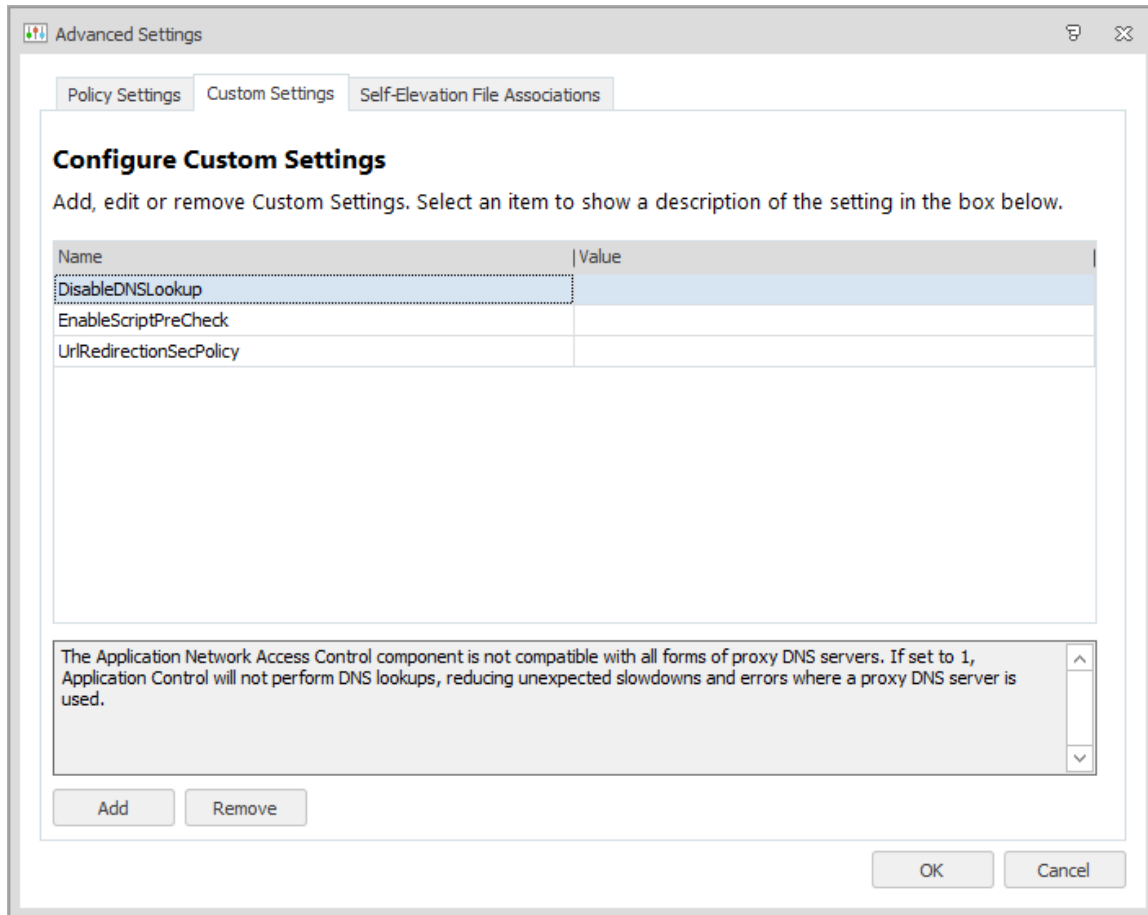
Functionality

Option	Description
Enable Application Access Control	Select to enable Application Access Control. Deselect to not validate or block executables.
Enable Application Network Access Control	Select to enable the Application Network Access Control feature. Deselect to not validate or block outbound network connections.
Enable User Privilege Management	Select to enable the User Privilege Management feature. Deselect to not apply any User Privilege policies. Disabling this option allows all applications to run with the permissions and privileges provided by default, by operating system. Application Control ignores anything in the User Privileges section of the rules and will not change or alter any of the user's privileges.
Enable URL Redirection	Select to enable the URL Redirection feature. If you deselect this option, configured redirections are ignored and users are not redirected when they enter a suspicious or unwanted URL. Deselecting this option has the same effect as having no items in the Browser Control policy set and selecting this feature. When you disable this feature the browser extensions for the Internet Explorer and Chrome browsers are disabled.

Signatures

Options	Description
Algorithm	<p>Select the algorithm type. There are three options available:</p> <ul style="list-style-type: none"> • SHA1 • SHA256 • Adler32 <p>For more information, see Signature Hashing.</p>

Custom Settings



Custom Settings allow you to configure additional settings which will be applied on managed endpoints when an Application Control configuration is deployed. If a new configuration is deployed that contains new custom settings, any pre-existing custom settings in place on the end point will be deleted.

Manage Custom Settings

1. Open a configuration in the Application Control Console and navigate to the Manage ribbon.
2. Click **Advanced Settings** and select the **Custom Settings** tab.
The Configure Advanced Settings dialog displays.
3. Select the Custom Settings tab and click **Add** to display the list of advanced settings.
4. Select the settings you want to configure and click **OK**.

- The selected settings are added to the Configure Advanced Settings dialog.



Settings which are added will be configured on the endpoint. However, any setting which already exists on an endpoint will be used.

- Set the values as required.
- Click **OK**.

The settings are applied when the configuration is deployed to your managed endpoints.

Available Custom Settings

Application Control contains the following configurable Custom Settings.

Setting	Data Type	Description
ADComputerGroupMembershipTimeoutSecs	Numeric	Timeout, in seconds, for nested computer group lookups. The default setting is 120 seconds and setting this value to 0 disables the timeout.
ADQueriesEnabled	Numeric	<p>This setting controls the types of AD queries used to determine the system's Distinguished Name and computer group membership.</p> <p>A value of 0 disables queries made to AD and the use of computer groups and OU in the configuration.</p> <p>The default value of 1 causes the agent to perform both the Distinguished Name and direct (non-nested) computer group AD queries. Nested computer groups in the configuration are ignored.</p> <p>A value of 2 causes the agent to perform the Distinguished Name, direct and nested computer group AD queries. This setting could cause performance issues on the DC due to high CPU usage.</p>
AlternateTOCheck	Numeric	Trusted Ownership checks have

Setting	Data Type	Description
		occasionally caused excessive CPU usage in the SYSTEM process when third party filter drivers are installed on the system. Enabling this setting, using a value of 1, causes Application Control to use an alternative method of looking up Trusted Ownership, which mitigates this issue in some cases.
AMFileSystemFilterFailSafe	Numeric	This setting configures whether the file system filter driver operates in a Fail Safe or Fail Secure mode. If there is a problem with the Agent and it stops responding, the driver disconnects in Fail Safe mode and does not intercept anymore requests. A value of 1 indicates Fail Safe, 0 indicates Fail Secure. Fail Safe is the default. Changing this setting requires an Agent restart to take effect.
AppHookDelayLoad	Text	This setting causes the AmAppHook DLL to load after a configurable number of milliseconds (ms) delay. This setting is configured on a per filename basis. The format is <filename+extension>,<delay>. The filename and extension can contain wildcards. Each pair is semi colon delimited. For example 'calc.exe,2000;note*.exe,6000'
AppHookEx	Text	Application Control utilizes a Windows hook as part of the Application Network Access Control (ANAC) feature. In rare cases, applications can display unexpected behavior when hooked. This setting is a list of applications in which ANAC specific functions are not hooked and therefore not subject to the ANAC rules.


Setting	Data Type	Description
		<p>If an application is named in both AppHookEx and UrmHookEx, the AmAppHook.dll is not loaded. Multiple entries are delimited by a semi-colon (;).</p>
AppInitDllPosition	Numeric	<p>Use this setting to specify whether the AsModLdr driver or the Appinit registry key is used to inject the Application Control hook. This setting is also used to determine the position of AMLdrAppinit.dll in the AppInit_DLL registry value.</p> <p>Set one of the following values:</p> <ul style="list-style-type: none"> • 0 - Positions the AMLdrAppInit.dll at the beginning of the AppInit_DLLs list. • 1 - Positions the AMLdrAppInit.dll at the end of the AppInit_DLLs list. • -1 - Excludes the AMLdrAppInit.dll from AppInit_DLLs and ASModLdr lists. When the AMLdrAppInit.dll is excluded from both lists, no automatic injection will occur. • 2 - Adds the AMLdrAppInit.dll to the ASModLdr list of dlls to be injected. This is the default setting. <p>This setting should only be used under the guidance of the Ivanti Support Team.</p>
baseconfigmergebehavior	Text	Use to control whether the new

Setting	Data Type	Description
		<p>configuration.aamp replaces or re-merges the existing merged_configuration.aamp. The accepted values for this setting are replace and remerge.</p> <p>When you merge using GPO the Replace value is ignored and automatically defaults to Remerge.</p>
BrowserAppStorePort	Numeric	Enter the port used to allow the Browser Control Chrome extension to be installed.
BrowserCommsPort	Numeric	Enter the port used for communications from browser extensions to the agent.
BrowserExtensionInstallHive	Numeric	<p>This engineering setting allows the administrator to choose which registry hive the Application Control Chrome browser extension will be installed in. Options are:</p> <ul style="list-style-type: none"> • 0 - Extension not installed • 1 - Install to HKLM • 2 - Install to HKCU. <p>0 is where the administrator must manually configure their own enterprise appstore to deploy the Application Control Chrome Extension. The default behaviour is 2 - for the chrome extension to be installed in HKCU.</p>
BrowserHookEx	Text	A semi-colon delimited list of applications that will not have the Application Control browser hook (BrowserHook.dll) injected. Applications in this list are not subject to URL Redirection, Web Installation or Elevated Website functionality.
BrowserNavigateEx	Text	A pipe () delimited list of navigation

Setting	Data Type	Description
		URLs that bypass the navigate event processing. The URLs in this list are not subject to URL redirection.
ComputerOUThrottle	Numeric	This setting limits an Active Directory look-up per connecting client for checking Organizational Unit membership by limiting the number of concurrent queries. This throttling helps reduce the amount of query-traffic on a domain if handling a large volume of connecting clients. Set this value between 0 and 65535.
ConfigFileProtection	Numeric	<p>Lock configuration AAMP files and the merged config folder to prevent configurations being updated by unauthorized users. This feature is disabled by default - set to a value to 1 to enable.</p> <p>Care should be taken when applying this setting in test environments - you may not be able to turn it off as your configuration cannot be updated. If this occurs, contact Ivanti Support.</p>
DFSLinkMatching	Numeric	DFS Link paths can be added to the rules. DFS Links and DFS Targets are treated as separate independent items to be matched. There is no conversion from Link to Target before applying the rules. Set this value to 1 to enable DFS Link matching.
DirectHookNames	Text	Application Control's Windows hook is loaded into all processes that load user32.dll by default. Applications which do not load this DLL are not hooked. Any applications which do not load user32.dll should be included in this setting as part of a semi-colon delimited

Setting	Data Type	Description
		list of full paths or filenames.
DisableAppV5AppCheck	Numeric	By default, any application launched using AppV5 is exempt from Trusted Ownership checking. Use this setting to disable this behavior with a value of 1.
DisableCustomRulesPreCheck	Numeric	This setting improves the performance of Custom Rules checking by only processing items that are configured within the policies of each custom rule collections. By default this setting is Off and set to '0'. Set the value to '1' to allow all potential requests through the custom rules.
DisableDNSLookup	Numeric	The Application Network Access Control (ANAC) component is not compatible with all forms of proxy DNS servers. If set to 1, Application Control will not perform DNS lookups, reducing unexpected slowdowns and errors where a proxy DNS server is used.
DisableSESecondDesktop	Numeric	By default, the auditing dialog for Self-Elevation displays on a second desktop. Set to 1 to display the dialog on the primary desktop.
DoNotWalkTree	Numeric	By default, process rules check the entire parent key for a match. This setting instructs process rules to only look at the direct parent of the process and not check the entire tree. A value of 1 enables this setting.
DriverHookEx	Text	A semi-colon delimited list of applications that will not have the Application Control Hook (AMAppHook.Dll) injected. Application Control requires the hook to be loaded for certain functionality to work. This


Setting	Data Type	Description
		custom setting should only be used under the guidance of the Ivanti Support Team.
EnableCustomRulesDllChecking	Numeric	By default this setting is off (set to 0) meaning only executables and URLs are processed. This setting improves the performance of Custom Rules checking by controlling whether DLLs are allowed through the rule collections. Set the value to 1 to allow all DLLs to be processed in addition to the default.
EnableScriptPreCheck	Numeric	<p>Whilst scripts within scripted rules are processing, they are treated as though they have returned a false value. The length of time scripts take, varies according to their content. This setting provides the best performance during computer start-up and user login because anything depending on the result of a script is not delayed. Set the value to 1 to make processes wait until the relevant script has finished. This can significantly slow down computer start-up and user login.</p> <p>Application Control does not wait indefinitely for scripts results - a 30 second timeout is applied.</p>
EnableSignatureOptimization	Numeric	<p>This setting improves the performance of rules checking, when using signatures. Files that do not match the full path are not hashed as it is assumed they are not the same file. Set to 1 to enable.</p> <p>Enabling this setting and ExtendedAuditInfo will not show any hashed file name in auditing metadata.</p>

Setting	Data Type	Description
ExplicitShellProgram	Text	This setting is used by Application Access Control (AAC). Application Control treats the launch of the shell program (by default explorer.exe) as the trigger for that session to be considered logged on. Different environments and technologies can change the shell application and the agent on occasion can't detect what the shell program is. Application Control uses the applications in this list (in addition to the default shell applications) to determine when a session is deemed to have logged on. This is a semi-colon delimited list of full paths or filenames.
ExProcessNames	Text	A list of space separated filenames that should be excluded from the filter driver. Changing this setting requires an Agent restart to take effect.
ExtendedAuditInfo	Numeric	This setting extends the file information for audited events. It reports the Secure Hash Algorithm 1 (SHA-1) hash, file size, file and product version, file description, vendor, company name, and product name for each file in its audited events. The information is added immediately after the file name in the event log. This setting is on by default. To turn it off, enter a value of 0. <div> The generation of a hash or checksum is disabled when the <i>EnableSignatureOptimization</i> setting is enabled.</div>
ImageHijackDetectionInclude	Text	A list of process names against which all

Setting	Data Type	Description
		child processes are verified to ensure the child image is running without corruption or modification and is a match for the one that was initially requested. If the child process is not verified, it is terminated. This is a semi-colon delimited list of full paths or file names.
MultipleHostsSameIP	Numeric	Allows Application Network Access Control (ANAC) to work with multiple hosts with the same IP Address. It takes out the caching of domain names to IP Addresses and allows different domains to work when running from the same server. Set to a value of 1 to enable.
NetEnableRevDNS	Numeric	<p>Used by Application Network Access Control (ANAC), this setting globally enables a reverse DNS lookup check on each request to access a network resource. Enabling this setting overrides the NetEnabledRevDNSList and RevDNSList settings. Set to a value of 1 to enable.</p> <p>This feature requires the administrator to enable and configure Reverse Lookup Zones on the company's DNS servers.</p>
NetEnableRevDNSList	Numeric	<p>Used by Application Network Access Control (ANAC), this setting enables a reverse DNS lookup check for only the IP addresses listed in the RevDNSList. This setting must be used in conjunction with the RevDNSList setting - set to a value of 1 to enable.</p>

Setting	Data Type	Description
		This feature requires the administrator to enable and configure Reverse Lookup Zones on the company's DNS servers.
OwnershipChange	Numeric	Application Control detects if a trusted file is changed by a non-trusted owner. In such a case, the file owner is changed to the untrusted user and any execute requests are blocked. Some applications overwrite files in such a way that Application Control does not detect it by default, therefore the owner of the file is not changed. When enabled, Application Control performs additional checks to catch all file changes and overwrites should be caught. Set to a value of 1 to enable.
PCRRetainOnNewConfig	Numeric	Control how Policy Change Requests (PCR) are dealt with when a new configuration is issued. This feature is disabled by default - authorized PCRs are removed upon receipt of a new configuration. Set a value of 1 to retain authorized Policy Change Requests when a new configuration is issued.
RdmHookEx	Text	A list of applications, used in Privilege Discovery Mode (PDM), in which PDM specific functions are not hooked by Application Control's Windows hook. The values should be a semi-colon delimited list of filenames.
RemoveDFSCheckOne	Numeric	When files are stored on a DFS drive, the Application Control agent uses a number of strategies to evaluate the correct UNC path. One of these strategies can cause delays during login

Setting	Data Type	Description
		if large numbers of scripts and executables are stored in and replicated by, Active Directory. Set to a value of one to enable, causing Application Control to ignore this strategy and increase performance in this situation.
RevDNSList	Varies	<p>This setting is only applicable when used in conjunction with NetEnableRevDNSList and is used by Application Network Access Control (ANAC). It contains IP addresses that will have a reverse DNS lookup check. The IP addresses should be in IPv4 dotted decimal format (n.n.n.n) and in a semi-colon delimited list.</p> <p>This setting requires the administrator to enable and configure Reverse Lookup Zones on the company's DNS servers.</p>
SECancelButtonText	Text	The text displayed by the cancel button on the Self-Elevation dialog.
SelfElevatePropertiesEnabled	Numeric	Set this value to '1' to enable self-elevation of properties. This feature is disabled by default.
SelfElevatePropertiesMenuText	Text	The text in the context menu option for self-elevation of properties.
SEOkButtonText	Text	The text displayed by the OK button on the Self-Elevation dialog.
TVChecking	Numeric	Enabling this setting causes Application Control to ignore Trusted Vendor checking for all files, even if the configuration contains entries for Trusted Vendors. Set to a value of 0 to enable this setting.

Setting	Data Type	Description
		This setting is Intended for troubleshooting issues.
UrlRedirectionSecPolicy	Numeric	<p>By default, the security policy is ignored by the URL Redirection feature. This engineering setting allows the administrator to force URL Redirection to follow the configured security policy. Set to a value of 1 to enable.</p> <hr/> <p> Self Authorization is not supported.</p> <hr/>
UrmForceMediumIntegrityLevel	Text	A User Privilege Management (UPM) custom setting used to override the integrity level when user privileges are elevated applications, which by default sets the integrity level to high. When this setting is used, the level is reduced to medium. This value should be a semi-colon delimited list of file names.
UrmHookEx	Text	<p>Application Control utilizes a Windows hook as part of the User Privilege Management feature. In rare cases, applications display unexpected behavior when hooked. This setting lists the applications where User Privilege Management specific functions are not hooked.</p> <p>If an application is named in both AppHookEx and UrmHookEx, the AmAppHook.dll is not loaded Multiple entries are delimited by a semi-colon.</p>
UrmPauseConsoleExit	Text	Used by the User Privilege Management feature. When a console application is elevated, a new application can appear in a new console window. The application runs to completion then

Setting	Data Type	Description
		closes. This is a problem if the user wants to see the output of the program. This setting causes the application to remain until a key is pressed. This is a semi-colon delimited list of full paths or filenames.
UrmSecPolicy	Numeric	By default, the security policy is mostly ignored by the User Privilege Management feature. User Privilege Management rules are applied in all cases except for when Audit Only mode is selected. This custom setting allows administrators to force User Privilege Management to follow the configured security policy. For Unrestricted and Self-Authorize security levels, User Privilege Management rules are not applied. For the Restricted level, User Privilege Management rules are applied. Set to a value of 1 to enable this setting.

Additional Engineering Key - GroupSidRefresh

Application Control requires the Security Identifier (SID) of all Group Rules to successfully perform rule matching. With this engineering key set, the agent will resolve the SID of the Group Rule at runtime whilst the endpoint is online and write it back into the Configuration (AAMP file). This can be useful if the endpoint is subsequently used offline as the SID stored in the configuration will be used.



The Application Control Console will resolve the SID if possible when the configuration is saved. This setting is only needed if the console could not perform the group SID lookup.

Settings

HKLM\Software\Ivanti Technologies\Application Control\Engineering

Name

GroupSidRefresh

Type

String (REG_SZ)

Parameters

0 - Off

1 - only resolve groups that currently have no SID values

2 - resolve all group SIDs –useful if the domain is specified by an environment variable so it is subject to change.

Self-Elevation File Associations

For further information, see [Self-Elevation File Associations](#).

Signature Hashing

To uniquely identify application files, a hash is taken of the file and stored in the configuration file. A hash is a unique digital signature for a configured application file and is generated using one of the following algorithms.

There are three supported algorithms:

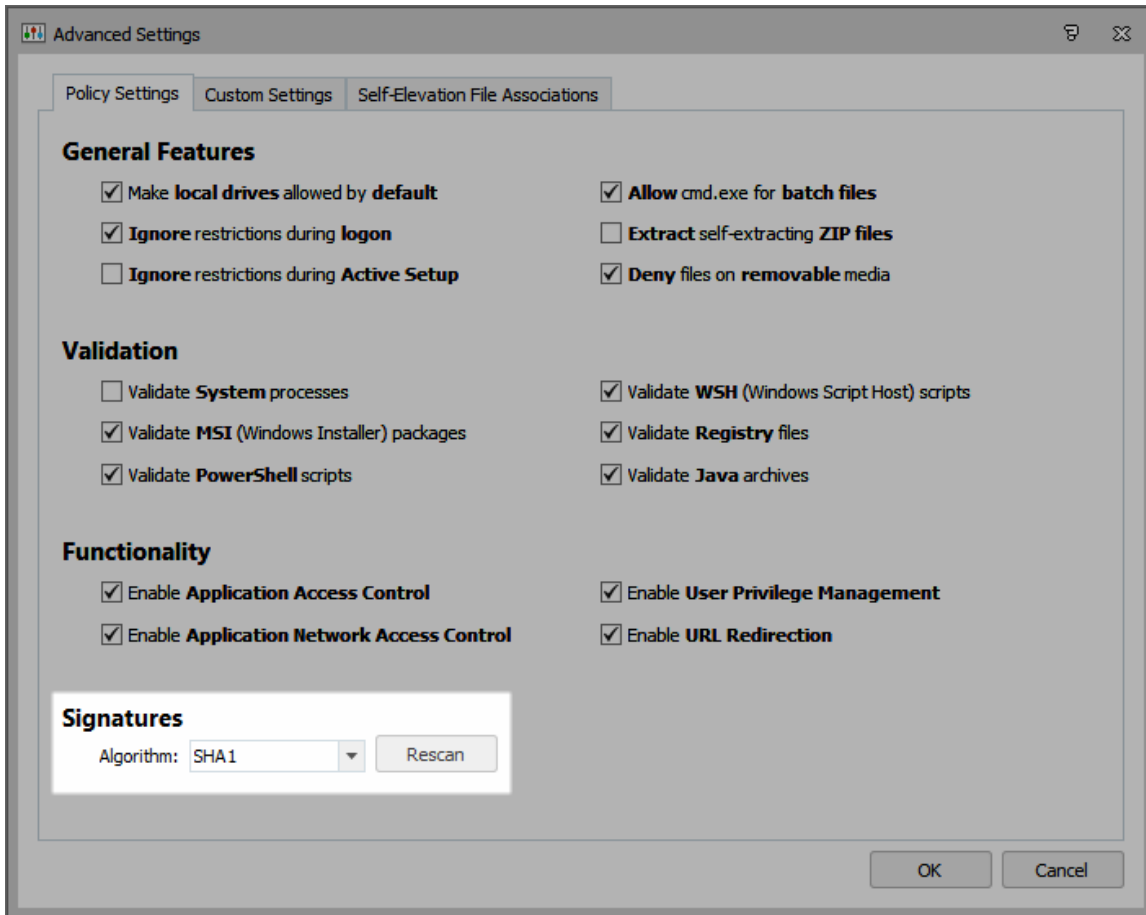
- **SHA-1** - The default hashing algorithm applied to each signature item in the configuration.
- **SHA-256** - A more complex, but slower hashing algorithm when compared with SHA-1.
- **Adler-32** - A lighter weight hashing algorithm when compared to SHA-1.

The hashing algorithm is a global option; this means only one can be set per configuration. This minimizes the amount of hashing that takes place on an endpoint running the configuration. If the hashing algorithm changes, or a file is updated, a rescan is required to generate a new hash code.



As it is a global setting, when Configuration Merging is being used, the algorithm in the base configuration is the one that takes precedent. For further information on Configuration Merging, see [Endpoint Configuration Merging](#).

Rescan File Signatures



When files are updated, for example after a Windows Update, you need to perform a rescan to align the hashing algorithm with the existing hash code.

1. On the Manage ribbon, in the General group, select **Advanced Settings**.
The Advanced Setting dialog displays.
2. To rescan all the updated files in a configuration, under Signatures, select the required algorithm and click **Rescan**.

3. The Signature Rescan dialog displays when the scan is complete.

The dialog contains three tabs:

- **Actions Required** - This tab displays when the signature rescan fails to find a previously hashed file or when a file path does not match the file stored in the configuration.

All missing files must be removed or manually located before the Signature Rescan dialog is closed.

- **Changed** - This tab contains an overview of all the application files that the new hashing algorithm has been applied to and their associated paths.
- **Unchanged** - This tab contains an overview of all the application files that already have the selected algorithm and have therefore not been changed.

4. If the rescan finds missing files, do one of the following:

- To delete the missing file from the configuration, select the filename from the Actions Required tab and click **Remove from configuration**.
- To locate the missing file manually, click the ellipsis, adjacent to the missing file and navigate to the file location.

Click **Export List** to produce a full report, in CSV format, that provides details of any files that are missing, changed or remain unchanged. The report provides details of the associated Rule Name, File, Hash and Status of all the hashed signature items. The exported report file can be opened in a spreadsheet so the data can be examined and queries run.

5. Click **OK**.

Any missing files must be removed or manually located before clicking **OK**.

The new hashing algorithm is applied and saved to all updated files in the Application Control configuration.

Apply a New Hashing Algorithm

Hashing algorithms can be applied to all files listed in an Application Control Configuration to help improve performance or to comply with localized rules and regulations. The following procedure shows you how to apply a new algorithm.

1. In the Manage ribbon, in the General group, select **Advanced Settings**.

The Advanced Setting dialog displays.

2. To change the hashing algorithm for all files in a configuration, select the algorithm type from the Algorithm drop-down. When a new algorithm is selected from the drop-down, a rescan of all files in the configuration is automatically triggered.

The Signature Rescan dialog displays when the rehashing is complete. This may take a few minutes, depending on the number of items being processed and the type of algorithm being applied.

The dialog contains three tabs:

- **Actions Required** - This tab displays when the signature rescan fails to find a previously hashed file or when a file path does not match the file stored in the configuration.

All missing files must be removed or manually located before the Signature Rescan dialog is closed.
 - **Changed** - This tab contains an overview of all the application files that the new hashing algorithm has been applied to and their associated paths.
 - **Unchanged** - This tab contains an overview of all the application files that already have the selected algorithm and have therefore not been changed.
3. If the rescan finds missing files, do one of the following:
 - To delete the missing file from the configuration, select the filename from the Actions Required tab and click **Remove from configuration**.
 - To locate the missing file manually, click the ellipsis, adjacent to the missing file and navigate to the file location.

Click **Export List** to produce a full report, in CSV format, that provides details of any files that are missing, changed or remain unchanged.

The report provides details of the associated Rule Name, File, Hash and Status of all the hashed signature items. The exported report file can be opened in a spreadsheet so the data can be examined and queries run.

4. Click **OK**.

Any missing files must be removed or manually located before clicking **OK**.

The new hashing algorithm is applied and saved to the Application Control configuration.

Auditing

The Application Control Auditing feature allows you to define rules for the capture of auditing information and to raise events, and includes a filter for specifying the events you wish to capture in the log. Auditing is accessed from the Manage ribbon.

Control general Auditing behavior

Send events to the **Application** event log ☐ Yes ☒ No

Send events to the **AppSense** event log ☐ Yes ☒ No

Make events **anonymous** ☐ Yes ☒ No

Send events to the **local** file log ☐ Yes ☒ No

Local log file **format** ☒ XML ☐ CSV

Select events to raise

Check the events you want to raise locally. Central event filtering defined with the Ivanti Management Console takes priority over this selection in a centralized deployment scenario.

[Toggle selected](#)

ID	Name	Description	Log Locally
9000	Denied execution	Denied execution request.	<input checked="" type="checkbox"/>
9001	Allowed execution	Allowed execution request.	<input type="checkbox"/>
9002	Overwrite changed owner	Overwrite of an allowed executable.	<input checked="" type="checkbox"/>
9003	Rename changed owner	Rename of a denied executable.	<input checked="" type="checkbox"/>
9004	Application limit denial	Application limit denial.	<input type="checkbox"/>
9005	Time limit denial	Time limit denial.	<input type="checkbox"/>

Event Filtering

OK Cancel

Control General Auditing Behavior

Use the following options to control the general auditing behavior and select the events to be raised:

- **Send events to the Application Event Log** - Select whether to send events to the Application Event Log.
- **Send events to the AppSense Event Log** - Select whether to send events to the AppSense Event Log. You can only send the events to the Application Event Log or the AppSense Event Log.
- **Make events anonymous** - Specify whether events are to be anonymous. If, Yes, the computer name and user name is omitted from all events. Anonymous logging also searches the file path for any instances where a directory matches the username and replaces the directory name with the string

- **Send events to local file log** - Select whether to send events to the local file log. If Yes, the events are sent to the local log file specified in the Text box. The default location is:
%SYSTEMDRIVE%\AppSenseLogs\Auditing\ApplicationManagerEvents_%COMPUTERNAME%
- **Local file log format**- Specify whether the event log is to be saved in XML format or CSV format.

In Enterprise installations, events can be forwarded to the Management Center via the Deployment Agent (CCA). When using this method for auditing, event data storage and filtering is configured through the Management Center console.



For more information, see the [Management Center Help](#).

Select Events to Raise

This section of the dialog lists all Application Control events. Select the **Log Locally** checkbox for the events you want to raise locally, in accordance with the selected auditing behavior options.

Available Events

Event ID	Event Name	Event Description
9000	Denied Execution	Denied execution request.
9001	Allowed Execution	Allowed execution request. A single request for an application can generate multiple 9001 events due to the way in which Windows responds to execution requests. So it's good practice to use event 9015 to accurately audit how many times a user has run an application.
9002	Overwrite Changed Owner	Overwrite of an allowed executable.
9003	Rename Changed Owner	Rename of a denied executable.
9004	Application Limit Denial	Application limit denial.
9005	Time Limit Denial	Time limit denial.
9006	Self-Authorization	Self-authorization decision by user.
9007	Self-Authorized allow	Self-authorization execution request.
9009	Scripted Rule Timeout	Script execution timed out.
9010	Scripted Rule Fail	Script failed to complete. This event is only raised for VB script failures.

Event ID	Event Name	Event Description
9011	Scripted Rule Success	Script completed successfully.
9012	Trusted Vendor Denial	Digital Certificate failed Trusted Vendor check.
9013	Network Item denied	Denied Network Item request.
9014	Network Item allowed	Allowed Network Item request.
9015	Application Started	<p>An allowed application started running.</p> <p>A single request for an application can generate multiple 9001 events due to the way in which Windows responds to execution requests. So it's good practice to use event 9015 to accurately audit how many times a user has run an application.</p>
9016	Unable to change ownership	The file's ownership could not be changed.
9017	Application Termination	A denied application has been terminated by Application Control.
9018	Application User Privileges Changed	The application's user privileges have changed.
9019	Web Installation allowed	Allowed Web Installation request.
9020	Web Installation restricted	Restricted Web Installation request.
9021	Web Installation restricted	Windows Restricted Web Installation request.
9022	Web Installation fail	Web Installation failed to complete.
9023	Self-Elevation allowed	Self-Elevation request.
9024	URL Redirection	URL Redirection has occurred.
9051	Policy Change granted	A Policy Change Request has been granted
9052	Policy Change invalid response code	An invalid response code has been entered for a Policy Change Request
9053	User-requested allow	An allowed Policy Change application has started
9054	User-requested elevate	An elevated Policy Change application has started
9055	Service start/stop	A service has been started or stopped.

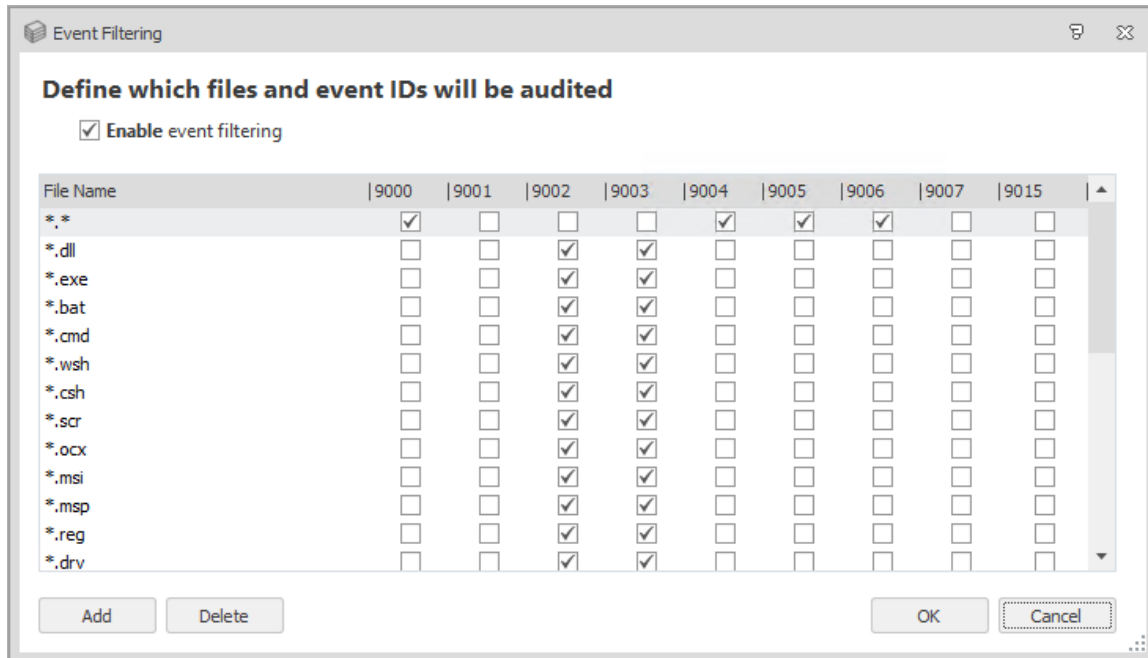
Event ID	Event Name	Event Description
9056	Untrusted file with metadata match	Failed to verify the certificate of a signed file when matching metadata
9096	Configuration merge success	The configuration merge has completed successfully.
9097	Configuration merge fail	The configuration merge has failed.
9098	Configuration merge timeout	The configuration merge has timed out waiting for expected files.
9099	Agent not licensed	Application Control is not licensed.

A single request for an application can generate multiple 9001 events due to the way in which Windows responds to execution requests. So it's good practice to use event 9015 to accurately audit how many times a user has run an application.



9001, 9007, 9014 and 9015 events are disabled by default as they can generate excessive event data on busy endpoints. We recommend these events are only used for troubleshooting purposes, and only for short periods of time.

Event Filtering



Event Filtering allows you to filter the file types that you want to audit. This is particularly useful if you choose a high volume event. The Event filter table is accessed by clicking **Event Filtering** in the Auditing dialog. The **Enable event filtering** is enabled by default and configured to ally the recommended file filters. Update the settings as required, selecting the file types to audit for each listed event. Click **Add** to specify new file types for the required event types.

System Events

The following are non-configurable system events:

Event ID	Event Name	Event Description
8000	Service Started	Application Control Agent: Service Started.
8001	Service Stopped	Application Control Agent: Service stopped.
8095	No Configuration found	Application Control cannot find a valid configuration.
8099	Invalid License	Application Control software is not licensed.

Configuration Profiler

The Configuration Profiler allows you to create a full report based on your current configuration or a report that focuses on a specific configuration items that match defined criteria such as the File, Folder, Network Connection, User, Group, and Device rule items. A full report also contains any conditions set for custom rules. Reports can be created whether configurations are stored locally or in a central database.

Use general reports to assist auditing and compliance requirements such as Sarbanes

When you create a Configuration Profiler report, the configuration must be loaded into the Application Control console. It does not need to be deployed.

Create a report with Configuration Profiler

1. In the Manage ribbon, click **Configuration Profiler**.
2. The Configuration Profiler dialog displays.
3. Select the report type.
4. If required define the criteria for the report.
5. Click **Create**.

Report Types

Select one of the following types of report:

- Complete Report - Produces a report which includes all aspects of the configuration, including any conditions set for custom rules.
- Report based on specific criteria - Produces a report which is based on the specified criteria as selected in the Report Criteria section.

Report Criteria

Report criteria are used when you want to create a report that focuses on specific configuration. In the Define Criteria section of the dialog, select from the following:

- User
- Group
- File
- Folder
- Network Connection
- Device
- Enter value to match - Enter the value to match for the associated criteria.

Report Output

When a report is created, the report is automatically displayed in a preview window where you can change the following:

- Paper
- Size
- Watermarks

When the changes have been applied, you are given the option to save the report in various formats, such as, PDF and Print.

Configuration Change Tracking

When Change Tracking is enabled, Application Control records any activity that occurs in the configuration. The information is stored in the Application Control package (AAMP) configuration file.

Configuration changes that are recorded include adding and removing User Groups, User Privilege Policies and changes to Group Rules.



Configurations generated through the scripting interface are not subject to configuration change tracking.

Enable or Disable Configuration Change Tracking

Change tracking is disabled by default for a new configuration. When a configuration is saved, so is the setting which becomes the default position.

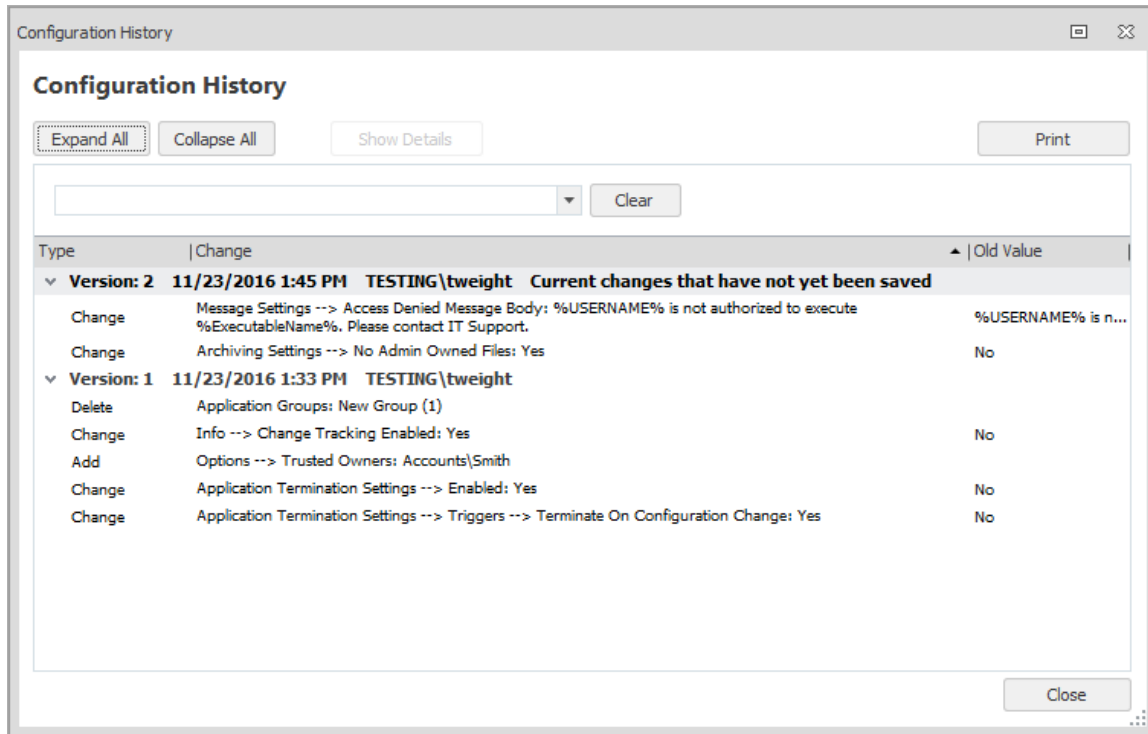
From the Manage tab, select **Enable Change Tracking** and enter your password. You have the option to password protect the feature when you initially click the Enable Change Tracking button. The password is used to prevent unauthorized users from performing task such as enabling and disabling the feature as well as deleting any Configuration Change Tracking history.

Once enabled, details of each change to the configuration are saved in the history and versioning is enabled.

When enabled, to stop recording configuration history, select **Disable Change Tracking** and enter your password. When Change Tracking is disabled, the history remains but no further changes are recorded.

If you disable Change Tracking and make changes to a configuration, when re-enabled, the configuration history shows that changes have been made whilst change tracking was disabled. It will not show any details of what has changed.

Configuration Change Tracking History



From the Manage ribbon, select **Configuration History** to display details of all the changes made to a configuration while change tracking is enabled. Whenever a configuration is saved, a new version of the history is created, outlining the changes made since the last save.

The Configuration History shows the following information for each change:

- **Type** - The change version and any actions that have been performed, for example, if you have enabled the Change Tracking feature, the Change action type would be logged in the history.
- **Change** - An overview of the change, for example **Info --> Change Tracking Enabled: Yes**. Double-click any entry in the history to access more details about a change.
- **Old Value** - Provides a brief description of what the original configuration was before any changes were applied.

Configuration Change Details

The screenshot shows a 'Change Overview' dialog box with the following fields and values:

User	TESTING\tweight
Date	11/23/2016 1:49 PM
Version	2
Change	Message Settings --> Access Denied Message Body: %USERNAME% is not authorized
Details	Message Settings --> Access Denied Message Body: %USERNAME% is not authorized to execute %ExecutableName%. Please contact IT Support.
Old Value	%USERNAME% is not authorized to execute %ExecutableName%

At the bottom of the dialog are three buttons: 'Previous', 'Next', and 'Close'.

Access more detailed information about each change by selecting the entry and clicking the **Show Details** button or double-clicking any history item from the following areas:

- Configuration History
- Review Changes dialog when saving a configuration

The Change field displays a high level overview, for example, "Change Tracking has been enabled". This is the same text that appears in the Configuration History dialog. The Details field provides more detailed information about the change. For example, if a line in a scripted rule is changed, the change history will display information on what the line was and what it has been changed to.

Export Change Tracking History

Configuration History can be exported to a CSV file. You can export the whole history of the configuration since change tracking was enabled or you can choose to export the history up to a certain date or configuration version.

By creating a backup, you can delete all or part of the history to reduce the configuration file size whilst ensuring that you still have access to the change tracking data. The exported history file can be opened in a spreadsheet so the data can be examined and queries run.

1. From the Manage tab, select **Export History**.
2. Select and configure the history you want to delete:
 - **All History** - Export the entire configuration history.
 - **History older than date** - Export the configuration history up to the entered date.
 - **History up to and including selected version** - Export the configuration history up to the specified version number.
3. Click **OK**.
4. Select a location to save the CSV file and click **Save**.

Delete Change Tracking History

Change history can be deleted when required and the amount of history you delete can be defined by date or version number. You are given the option to export the history prior to deleting.

1. From the Manage tab, select **Delete History**. If you specified a password when enabling Configuration Change Tracking, you will be prompted for this password.
2. Select and configure the history you want to delete:
 - **All History** - Delete the entire configuration history.
 - **History older than date** - Delete the configuration history up to the entered date.
 - **History up to and including selected version** - Delete the configuration history up to the specified version number.
3. Click **OK** and select whether you want to **Export then Delete** or just **Delete** the history. If you export prior to the delete the selected history is exported to CSV file at a selected location.

Deleting the history does not change or remove version numbers. When the history is deleted, the version numbers stay the same but increment as normal on future saves.

Undo and Redo Changes



If you undo a configuration change using the buttons in the quick access menu, the history of that change is removed. If an undone change is redone, the history is restored.

Save a Configuration

When you save a configuration to disk, the Management Center, Group Policy, SCCM or as the live configuration on an endpoint, an overview of the changes you have made since the last save is displayed.

Each time a configuration is saved, its version number is incrementally increased and displayed at the bottom right of the console - regardless of whether change tracking is enabled or not.

Privilege Discovery Mode

The Privilege Discovery Mode is accessed from the Configuration navigation button, Privilege Discovery Mode node and provides the functionality to monitor endpoints in order to identify applications that use administrative privileges. A web service is used to collect the data and relay that data to the Privilege Discovery Results work area in the Application Control Console. The data listed in the reports can be used to simplify the creation of an appropriate Application Control configuration and to produce reports.



Privilege Discovery Mode is intended for use during a discovery or pilot phase, so a maximum of 500 endpoints is recommended, depending on hardware specifications.

Web Services

Application Manager Web Services are installed on any selected machine as part of the Application Control installation. It is a lightweight component that does not require typical server tools such as IIS or SQL Server. Although Application Manager Web Services installs without any need to configure it, the default configuration can be amended using HttpCfg or Netsh tools. When installed, the Service runs in the background when Privilege Discovery is configured and monitors client endpoint activity tracking details such as the applications that use administrative rights, the names of users using the application, and the name of the endpoint it was launched from.

The results of the tracking are displayed in the Applications Manager Console using the Privilege Discovery Results work areas and they can be used to generate reports and create Application Control configurations.

For more information, see [Web Services Configuration](#).

Configure Privilege Discovery

Privileges Discovery is configured using the Privileges Discovery Mode node accessed from the Configuration button in the navigation pane and is activated by selecting Enable Privileges Discovery Mode.

The Privileges Discovery Mode work area contains:

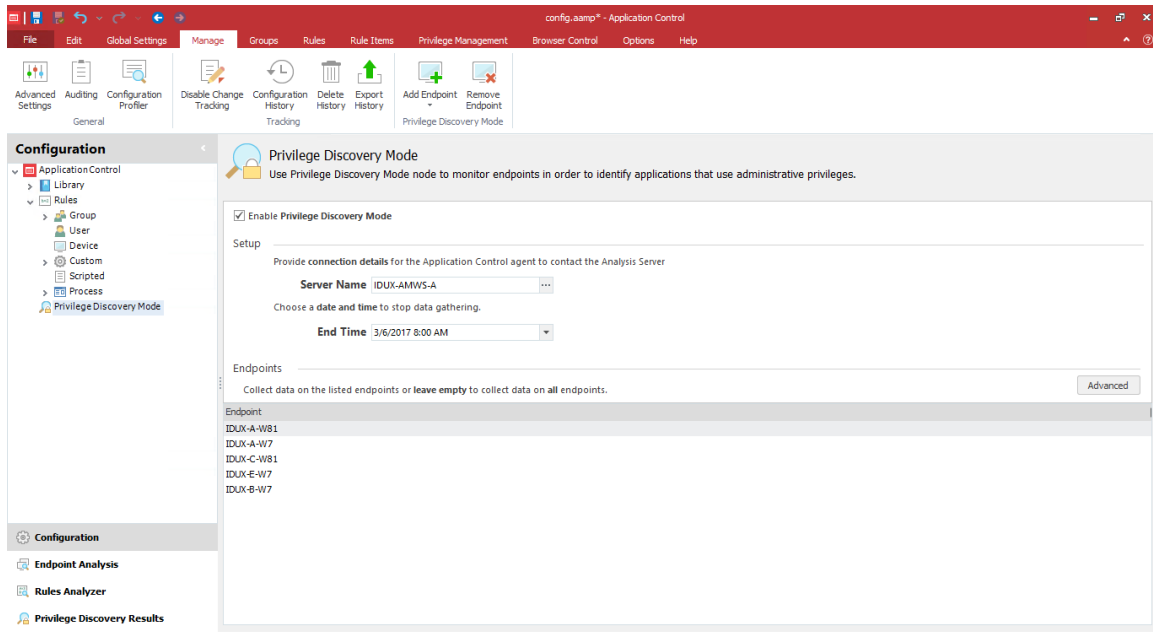
- **Setup** - Use the Setup area to determine the server name and location for the Application Control Agent to contact the Service. You can also choose when data collection is to finish by selecting the date and time from the End Time field.

It is recommended that the time period is set far enough ahead to maximise the number of applications captured and therefore, improve control of administrative rights used on your network.

- **Endpoints** - Allows you to specify the endpoints from which the data is collected. To specify endpoints from individual deployment groups or work groups, right-click in the Endpoint area and select **Add Endpoint**.
- **Advanced Button** - Use the Advanced button to configure the Privileges Discovery advanced features. These include configuring the communication port to be used by the Privileges Discovery Mode and the frequency by which the collected data is fed back to the Application Manager Web Service.

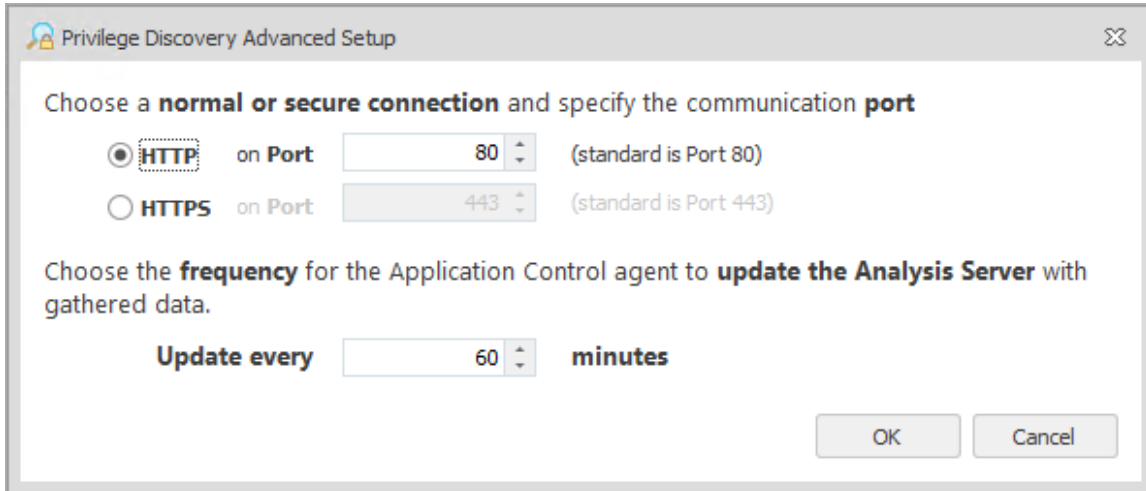
The Privilege Discovery Mode ribbon allows you to add or remove endpoints when the Privilege Discovery Mode node is selected in the Configuration navigation pane. Use the Add Endpoint button to specify an endpoint to collect data from. The Remove Endpoints option provides you with the facility to remove a highlighted endpoint so that it will no longer be monitored.

Configure Privilege Discovery Mode



1. Select the **Configuration** navigation button.
2. Select the **Privilege Discovery Mode** node.
3. In the work area, select **Enable Privilege Discovery Mode**.
The Privilege Discovery Mode options becomes available.
4. In the Server name field, select the ellipsis (...) to browse for the Application Control Web Server to be used. The name of the server can also be entered manually into the field.
5. In the End Time field, specify the date and time that the server will stop gathering application information.
6. To specify particular endpoints to be monitored, right-click in the Privilege Discovery work area and do one of the following:
 - Select **Browse Deployment Group** to locate the deployment group to be monitored.
 - Select **Browse Domain/Workgroup** to locate the domain or specific workgroup to be monitored. If no endpoints are added to the work area, data will be collected from every configured endpoint.
7. If required, advanced settings can be configured using the **Advanced** button.
8. Save the configuration.

Configure Privilege Discovery Advanced Settings



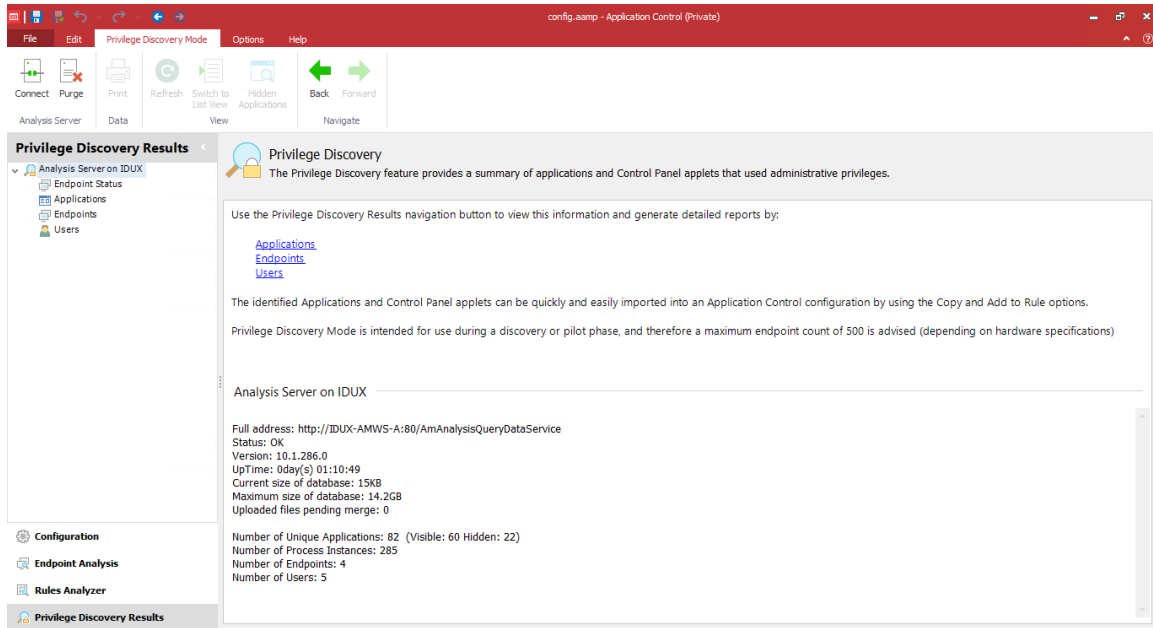
The dialog box is titled "Privilege Discovery Advanced Setup". It contains two main sections. The first section is titled "Choose a normal or secure connection and specify the communication port". It has two radio buttons: "HTTP" (selected) and "HTTPS". Next to each radio button is a text label "on Port" followed by a spin box. For HTTP, the spin box shows "80" and has a note "(standard is Port 80)". For HTTPS, the spin box shows "443" and has a note "(standard is Port 443)". The second section is titled "Choose the frequency for the Application Control agent to update the Analysis Server with gathered data." It has a text label "Update every" followed by a spin box showing "60" and the word "minutes". At the bottom right are "OK" and "Cancel" buttons.

Advanced settings are optional and allow you to configure Privilege Discovery further by providing the facility to specify the types of connection and the specific communication ports. You can also choose how often the Application Control Agent updates the Analysis Server with the gathered data by entering the time in minutes.

1. In the Privilege Discovery Mode work area, select the **Advanced** button.
The Privilege Discovery Advanced Setup dialog displays.
2. Select one of the following options:
 - HTTP - Select this to use the standard application protocol and enter the port number you require.
 - HTTPS - Select this to use the secure application protocol and enter the port number you require.
3. To amend the time by which the agent is to update the Application Manager Web Server, enter or select the time in the Update Every field. The default setting is 60 minutes.
4. Click **OK**.

Begin gathering the privilege discovery information by deploying the configuration to each of your endpoints.

Privilege Discovery Results



The results from the Privilege Discovery are viewed using the Privilege Discovery Results navigation button accessed from the Application Control console. The results are separated into the following nodes:

- **Applications** - The Applications node opens the Application Summary work area and provides access to Application Details. The information in the Application Summary page can be viewed by application icon or as a list by clicking the Switch to List View button and provides details of applications that used administrative privileges to run.

Further information such as the endpoint on which the application was run, the name of the user, the command line that was used to execute the application and the time it was launched, can be accessed when you double-click on a specific application in the summary work area.

Application Details can also be used to create an Application Control configuration that can then be applied to user privileges for specific groups or users.

- **Endpoints** - The Endpoints node opens the Endpoint Summary work area and provides access to Endpoint Details. The Endpoint Summary page displays the name of the endpoint, the number of unique applications run from the endpoint, the number of users with administrative privileges that used that endpoint and the how many times a particular application was run.

Further information such as user details, name of the applications used, the command line that was used to execute the application and the time applications on the endpoint were launched can be accessed when you double-click on a specific endpoint in the summary work area.

Endpoint details can also be used to create an Application Control configuration that can then be applied to user privileges for specific groups or users.

- **Users** - The Users node opens the User Summary work area and provides access to User Details work area. The information in the Users Summary page consists of the username, the number of times unique applications were run, how many endpoints a particular user accessed and how many instances of an application were run.

Further information such as user details, the number of times a unique application was run with Administrative privileges, the endpoints used and the number of instances applications were run are displayed.

The User Details can also be used to create an Application Control configuration that can then be applied to user privileges for specific groups or users.

Each of the nodes can be used to create user rules for specific groups or users based on the results you have selected. They are then added to an Application Control configuration to be distributed to the endpoints on your network.

The server details are also accessed from the Privilege Discovery Results navigation tree. The details displayed allow you to keep a track of the endpoints being monitored together with details of when monitoring started and the predicted time of completion.

Add a User Rule from the User Results

When the Privilege Discovery monitoring period elapses, the results are collected on the Application Control Web Server and can be viewed using the Privilege Discovery Results navigation button. These results can be used to create rules that be included and distributed to endpoints on your network.

1. Click the **Privilege Discovery Results** navigation button and select **Users**.
2. Do one of the following:
 - To view detailed information about the selected user, go to Step 3.
 - To add an application or control panel applet used by a user straight to a rule, go to Step 5.
3. Double-click the selected user to display the information.

The User Details work area displays

4. Expand and collapse the nodes as required, to access further information.
5. Right-click on the application or control panel applet to be added to a user rule.
6. Select **Add to Rule** and specify where the rule should be added.
7. Do one of the following:
 - Select **as file name** to add as a file name to the designated User Privileges node.
 - Select **as signature** to add as a signature to the designated User Privileges node.
 - Select **as full command line** to add command line control to the designated User Privileges node.

You can check that the application or control panel applet has been added to the correct node by navigating to it in the Configuration navigation tree.

8. Save the configuration.

Add a User Rule from the Endpoint Results

When the Privilege Discovery monitoring period is underway, you can add application items, specific endpoints or associated user components directly to an Application Control configuration and then distribute the configuration file to all the configured endpoints on your network.

It is recommended that, where possible, you wait for the Privilege Discovery period to elapse in order to create the configuration file.

1. Click the **Privilege Discovery Results** button in the navigation pane and select **Endpoints**.
2. Do one of the following:
 - To view detailed information about the selected endpoint, go to Step 3.
 - To add an application or control panel applet used by an endpoint straight to a rule, go to Step 5.
3. Double-click the selected endpoint to display the information.

The Endpoint Details work area displays

4. Expand and collapse the nodes as required to access further information.
5. Right-click on the application or control panel applet to be added to a user rule.
6. Select **Add to Rule** and specify where the rule should be added.

7. Do one of the following:
 - Select **as file name** to add as a file name to the designated User Privileges node.
 - Select **as signature** to add as a signature to the designated User Privileges node.
 - Select **as full command line** to add command line control to the designated User Privileges node.

You can check that the application or control panel applet has been added to the correct node by navigating to it in the Configuration navigation tree.

8. Save the configuration.

Add Discovered Applications to a User Rule

When the privilege discovery monitoring period is underway, you can add application items, specific endpoints or associated user components directly to an Application Control configuration and then distribute the configuration file to all the endpoints on your network.

It is recommended that, where possible, you wait for the Privilege Discovery period to elapse before creating the configuration file.

1. Click the **Privilege Discovery Results** Navigation button and select **Applications**.
2. Do one of the following:
 - To view detailed information about the selected application, go to Step 3.
 - To add the application or control panel applet straight to a rule, go to Step 5.
3. Double-click the selected application to display the information

The Application Details work area displays
4. Expand and collapse the nodes as required, to access further information.
5. Right-click on the application or control panel applet to be added to a user rule.
6. Select **Add to Rule** and specify where the rule should be added.
7. Do one of the following:
 - Select **as file name** to add as a file name to the designated User Privileges node.
 - Select **as signature** to add as a signature to the designated User Privileges node.
 - Select **as full command line** to add command line control to the designated User Privileges node.

You can check that the application or control panel applet has been added to the correct node by navigating to it in the Configuration navigation tree.

8. Save the configuration.

Add Known Applications to the Hidden Applications list

Once you have dealt with a discovered application, either by adding it to the configuration so that it will run elevated with administrative rights, or by deciding that end users should not be able to run the application with administrative rights, then you can hide the application from displaying in any of the report views. You do this by adding the known applications to an automatic exclusion list called "Hidden Applications". When applications are added to this list they are automatically ignored in any future Privilege Discovery Reports and are no longer displayed as part of your results. To add applications to this list, highlight one or more of the applications, right-click and select **Hide Application** from the context menu.

The Hidden Applications list is accessed using the Hidden Applications button in the Privilege Discovery Mode ribbon and allows you to view and restore previously excluded applications using the Restore and Restore All buttons.

Privilege Discovery Status

The server details are displayed when you first select the Privilege Discovery Results navigation button. The details include the server information, current and maximum allowable database size, and details of the endpoint usage.

The Endpoint status node provides details of the endpoints currently being monitored with information such as the date and time of the last update together with the time the privilege discovery report is scheduled to finish.

Group Management

Group Management is a library for compiling reusable groups of files, folders, drives, signatures and network connections that can be associated with rules in the configuration. For example, Groups can be used to manage licenses for a suite of software or common sets of applications for assigning to certain user groups.

Use Groups to help manage long lists of related items for an application, for example, all the File, Folder, Drive, Signature, Windows Store Apps, and Network Items. Add the groups to rules to allow or restrict access. Groups can include any combination of these items. For example, you can group a number of items for one particular application and then add the group to the Allowed or Denied Lists.

If the Group Name is amended, it automatically updates in any rule where the group is applied.

Create a Group

Two groups cannot have the same name. Naming two groups the same will display an error message informing that a group with the same name exists. You cannot save the group until you specify a unique name.



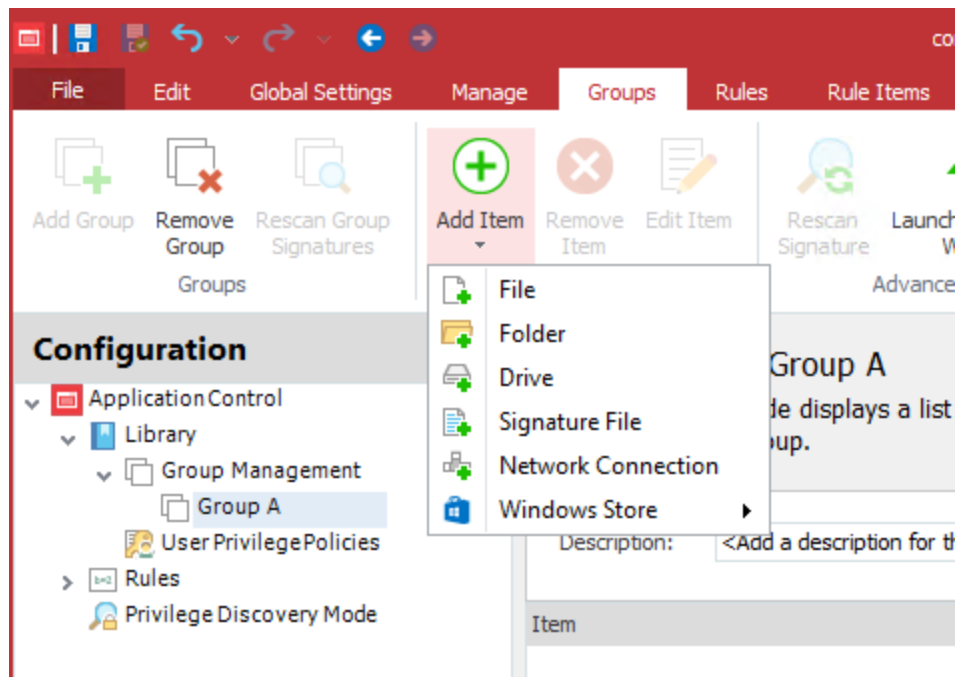
Renaming a group reflects in all rules that use that group.

1. Navigate to the Group Management node.
2. Select **Add Group** on the Groups ribbon.
3. The new group with the default name, New Group, is added below the Group Management node.
4. To rename the group, double-click it to make the name editable and enter a new meaningful name, for example, *Microsoft Applications*.
5. To sort the groups, right-click the **Group Management** node and select **Sort Ascending** or **Sort Descending**.

Add Items to a Group

1. Any combination of Files, Folders, Drives, Signature Files, Windows Store Apps and Network Connections can be added to a group. For example, all items that belong to a single application.
2. Navigate to the **Group Management** node and select the group you want to add items to.

3. Click the **Add Item** drop-down arrow on the **Groups** ribbon.



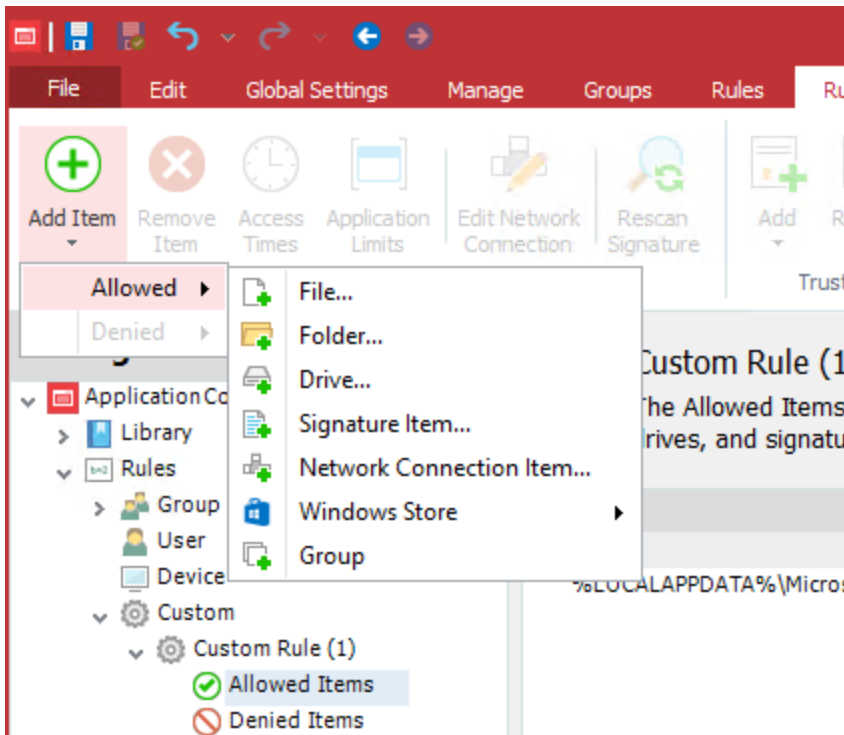
4. Do one or more of the following:
- To add a file, select **Add > File**
 - To add a folder, select **Add > Folder**
 - To add a drive, select **Add > Drive**
 - To add a signature file, select **Add > Signature File**
 - To Add a Network Connection item, select **Add > Network Connection Item**
 - To Add a Windows Store App, select **Add > Windows Store App**

To populate a group, you can also do the following:

- Right-Click a group and select **Add Items**.
- Add multiple files at once.
- Drag and drop items from Internet Explorer. Note dragging and dropping files also includes any dependencies.
- Cut, copy and paste between groups.

You cannot add duplicate items to a group.

Add Groups to a Rule Item



Groups can contain a number of items, for example, all the File, Folder, Drive, Signatures, Windows Store Apps and Network Items for a single application.

You can add groups to the [Allowed Items](#), [Denied Items](#), and [User Privileges](#) rule items, eliminating the need to add items individually to the lists.

1. Select either an **Allowed Items**, **Denied Items** or **User Privileges** rule item.

The rule item work area displays.

2. Click **Add Item**, and then select the menu path to add a group as follows:
 - To add an Allowed Item, select **Allowed > Group**.
 - To add a Denied Item, select **Denied > Group**.
 - To add a User Privileges Item, select either **Application > Group** or **Self-Elevation > Group**.

The group selection dialog displays.

3. Select the group you want to add and click **OK**.

The group is added to the rule.

Remove Groups from a Rule Item

You can remove a group from a rule. All items within the group are also removed from the rule item. The group is not deleted and still remains under the Group Management node.

1. Select the **Allowed Items**, **Denied Items** or **Privilege Management** rule item within the rule that contains the group you want to remove.

The work area displays.

2. Select the group you want to remove and select **Remove Item** in the Rule Items ribbon.

The Remove Items dialog box displays.

3. Click **Yes**.

The group is removed.

Delete a Group

You can delete a group. When a group is deleted all items within the group are also deleted. If you try to delete a group that is currently used by a rule, a dialog displays that tells you where the rule where the rule is used. Remove the group from the rule before you delete the group. A message is displayed when the group contains items such as File, Folder, Drive, Signatures, Windows Store Apps, and Network Items.

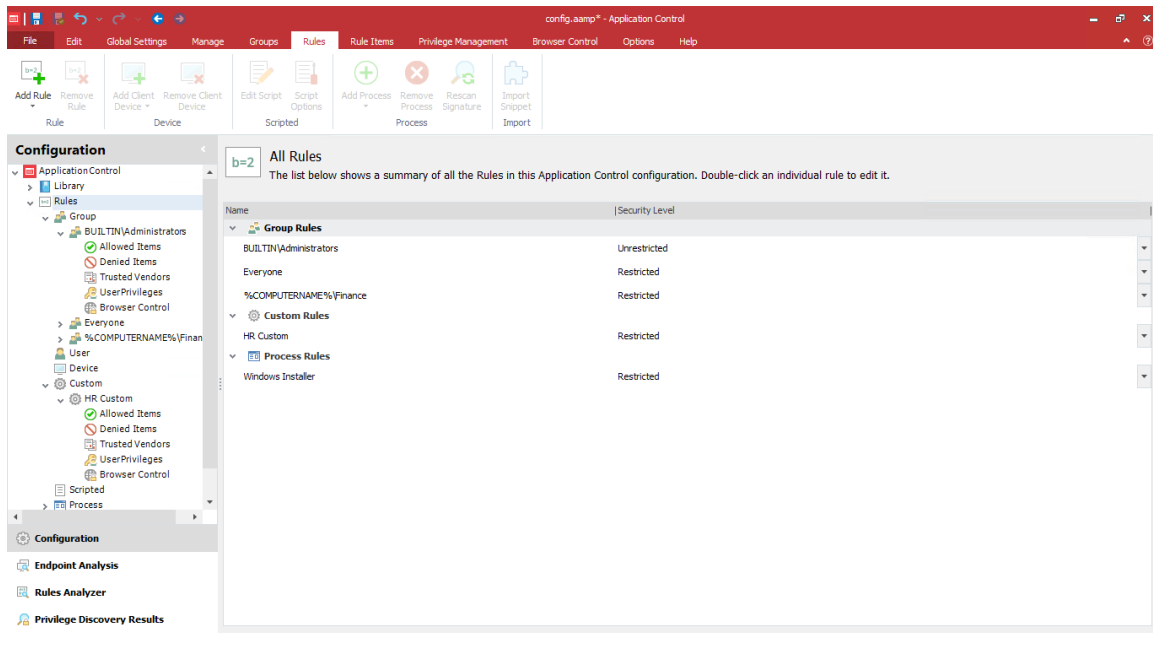
1. Select the group you want to delete.
2. Select **Remove Group** on the Groups ribbon.
3. One of the following occurs:
 - The Confirm Removal dialog displays. Click **Yes**. The group and the items it contains are deleted.
 - The Group in use dialog displays providing the location of the rules that reference the group. Click **OK** and remove the group from the rule. Select **Remove Group** on the Groups ribbon, and click **OK** in the Confirm Removal dialog.

Capture Signatures in a Group

Use the Signature Wizard to capture multiple signature files.

1. Select the group that you want to add signatures to.
2. Select **Launch Signature Wizard** on the Groups ribbon.
The Signature Wizard displays.
3. Click **Next**.
The Search method window displays.
4. Do one of the following:
 - To search for files in a particular folder, go to Step 5. If you wish to examine a specific process, make sure you have launched the relevant application before proceeding.
 - To examine files used by one of the processes running on the computer, go to Step 10.
5. Select **Search Folders** and click **Next**.
The Searching Folders window displays.
6. Browse to and select the folder you wish to search and click **OK**.
7. Select the **Include subfolders** option as required and click **Next** to begin the search.
The Review Files window displays.
8. Review the files and click **Next** to capture the signatures.
The Signature Generation window displays
9. Go to Step 14.
10. Select **Examine a running process**.
11. Click **Next**.
The Examine a running process window displays showing all the running processes.
12. Select the process to examine and click **Next**.
The Review Files window displays
13. Review the files and click **Next** to capture the signatures.
14. Allow the generation to complete then click **Next** and **Finish**.

Rules



Rule nodes allow you to create rules targeting specific users, groups, and devices, and assign security level policies, resource access, and resource restrictions that apply to the users, groups, and devices that match the rules. There are six rule types:

- [Groups Rules](#)
- [User Rules](#)
- [Device Rules](#)
- [Custom Rules](#)
- [Scripted Rules](#)
- [Process Rules](#)

Rule nodes provide Security Level settings for specifying the levels of restrictions to execute files. Application Control configuration rule settings security levels specify how to manage requests to run unauthorized applications by the users, groups, or devices that a rule matches:

- **Restricted** - Only authorized applications can run. These include files owned by members of the Trusted Owners list and files listed in Allowed Items, Trusted Vendors, and Trusted Applications.
- **Self-Authorizing** - Users are prompted for decisions about blocking or running unauthorized files on the host device.
- **Audit only** - All actions are permitted but events are logged and audited, for monitoring purposes.

- **Unrestricted** - All actions are permitted without event logging or auditing settings for specifying the levels of restrictions to execute files.

Rule nodes also provide a further layer of granularity for controlling application use with Allowed Items, Denied Items and Trusted Vendors for specifying lists of files, folders, drives and signature items, network connection items, Windows store apps, and groups that are allowed or prevented from running.

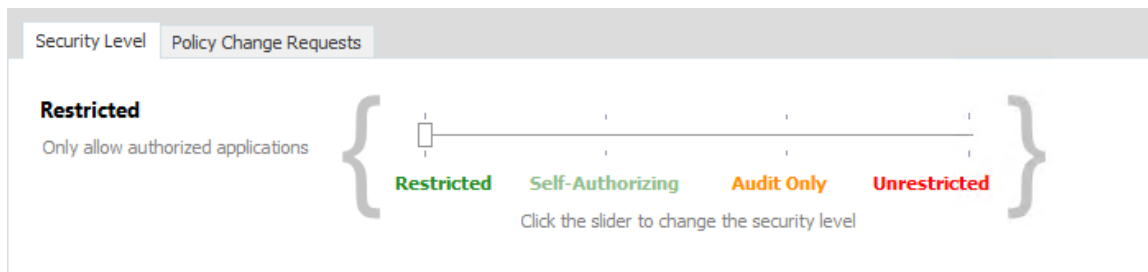
For more information for the relevant options for each Rule Type, see [Rule Options](#).

To display all rules in the configuration, click **Rules** in the navigation tree. A summary displays all rules listed under the relevant rule type. The security level assigned to each rule is seen and can also be amended.

Security Levels

Apply security levels to control whether the user, group, and devices specified in a rule are fully restricted by Application Control rules, unrestricted, audited only, or granted self-authorization status entitling the user to decide whether to run an application. Self-authorized users can be audited by raising events in the Auditing component and the Windows Event Log.

Set the Security Level



To set the Security Level, select the required node and, using the slider, apply the required security level.

Restricted

Select to restrict users, groups, and devices in the rule to run only authorized applications. These include files owned by members of the Trusted Owners list and files listed in the Allowed Items node.

Self-Authorize

Select to prompt users, groups, and devices in the rule to decide whether to allow execute requests for each unauthorized file. Unauthorized files either do not belong to the Trusted Owners list or are not specified in the Allowed Items list of a given rule.

A self-authorizing user prompt includes the following options:

- **Allow** - Allows the application to run.
- **Block** - Blocks the application from running.

When a DLL file is allowed to run, a message notifies the user that the application which uses the DLL may need to be restarted. The default message which displays can be modified in the Message Settings dialog on the Global Settings ribbon.

Users can also decide how long the setting is applied for:

- **Remember my decision for this session only** - The authorization decision is upheld only for the current session. The user is prompted again for an authorization decision when attempting to run an application in any future sessions.
- **Remember my decisions permanently** - The user decision is upheld for all future sessions.

If neither of these options are selected, the decision is upheld only for the current instance the user is attempting to run. The self-authorization prompt is reissued for any future attempts to run instances of the application.

Audit Only

Select to permit all actions but log and audit events for monitoring purposes, according to the policy settings in [Auditing](#).

Unrestricted

Select to permit all actions without even logging or auditing.

Example: Testing Self-Authorization

You can test whether Security Levels are being implemented correctly. The following example shows you how to test the Self-Authorizing level.

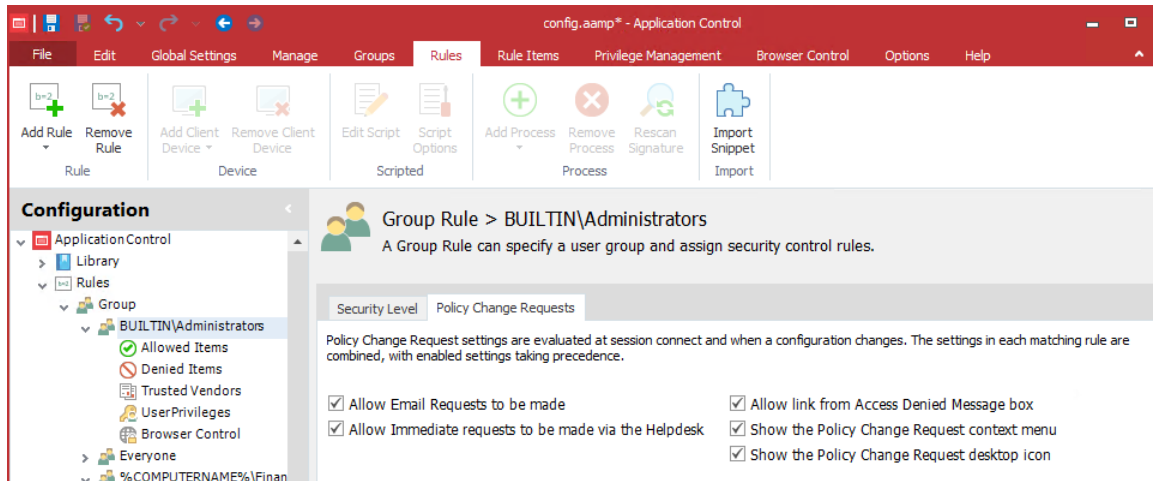
1. Create a rule in the User rules node that applies to a test user account that is not a member of a group that belongs to the Trusted Owners list.
2. Set the security control level to Self-Authorizing to allow the test user to self-authorize applications to run.
3. Save the configuration.
4. Run the Registry Editor. The application is prohibited and a message box displays with a prompt for a decision to allow the file to run and informing that the action will be logged.

Example: Change the Security Level for a Process Rule

You can apply security levels to control whether applications specified in the process rule are fully restricted by Application Control rules, unrestricted, or audited only.

1. Select the required process rule.
The **Process Rule** work area displays.
2. Click and drag the **Security Level** slider to the required level.

Policy Change Request Settings



Configure which request types and features are available to users for each rule. Policy Change Request settings are available for all rule types, apart from Process rules.

1. Select a rule in the navigation pane.
2. Select the **Policy Change Requests** tab.
3. Apply the required settings for the selected group:
 - Allow Email Requests to be made
 - Allow Immediate requests to be made via the Helpdesk
 - Allow link from AppSense Access Denied Message box
 - Show the Policy Change Request context menu
 - Show the Policy Change Request desktop icon

The detail for each setting is configured using the Policy Change Requests dialog, accessed from the Global Settings ribbon.

Group Rules

The Group rules node allows you to match security control rules with specific user groups within the enterprise.

The Group summary displays the group name, Textual Security Identifier (SID) and Security Level of the rule. Application Control allows you to assign four distinct security levels to the group rules. An SID is a data structure of variable length that identifies user, group, and computer accounts. Every account on a network is issued a unique SID when the account is first created. Internal processes in Windows refer to an accounts SID rather than the accounts user or group name. Likewise, Application Control also refers to a user or group SID unless the SID could not be found when added to the configuration.

There are two predefined Group rules:

- **BUILTIN\Administrators** - Users in BUILTIN\Administrators are assigned the Unrestricted security level. The BUILTIN\Administrators group is for managing access to the applications for local administrators.
- **Everyone** - The Everyone group rule and all additional group rules have a security level of Restricted, unless a user matches other group or user rules with higher priority settings. All users, including administrators are part of the Everyone group. This means administrators are part of two group rules: the BUILTIN\Administrators group, which is unrestricted, and the Everyone group, which is restricted. Application Control uses the least restrictive rules; therefore, all administrator requests are unrestricted.

Typically, you specify all the files, folders, drives, signature items, network connection items, and groups to prohibit for Everyone. You can then create a new group or user and specify the items you want to be accessible for that group or user. This enables you to control what users have access to.

Manage group rules as follows:

- To add a group rule, click the **Add Rule** on the Rules ribbon and select **Group Rule**.
The Add Group Rule dialog displays. Enter or browse to select an account.
- To remove a group rule, highlight a rule and click **Remove Rule** on the Rules ribbon.

A confirmation message displays. Click **Yes** to confirm the removal.

You can also add items to the Allowed Items, Denied Items, Trusted Vendors, User Privileges, and Browser Control nodes in each group rule node

For more information, see [Rule Items](#).

User Rules

The User rules node allows you to match security control rules with specific users within the enterprise.

The User summary displays the User, Textual Security Identifier (SID) and Security Level of the rule. An SID is a data structure of variable length that identifies user, group, and computer accounts. Every account on a network is issued a unique SID when the account is first created. Internal processes in Windows refer to an accounts SID rather than the accounts user or group name. Likewise, Application Control also refers to a user or group SID unless the SID could not be found when added to the configuration.

- To add a user rule, click the **Add Rule** on the Rules ribbon and select **User Rule**. The Add User Rule dialog displays. Enter or browse to select an account.
- To remove a user rule, select a rule and click **Remove Rule** on the Rules ribbon. A confirmation message displays. Click **Yes** to confirm the removal.

You can also add items to the Allowed Items, Denied Items, Trusted Vendors, User Privileges, and Browser Control nodes in each user rule node.

For more information, see [Rule Items](#).

Device Rules

Device rules allow security control rules to be matched with specific devices. Device rules can apply the rule settings either to the device hosting the Application Control Agent and configuration, or to connecting devices.

For example, a configuration rule can allow certain applications to run on a server but prohibit others from running when launched from a device listed in the rule.


Device rules also provide the ability to perform per-device license management in a server-based computing environment.

- To add a device rule, click **Add Rule** on the Rules ribbon and select **Device Rule**.
- To remove a device rule, select a rule and click **Remove Rule** on the Rules ribbon. A confirmation message displays, click **Yes** to confirm the removal.

You can also add items to the Allowed Items, Denied Items, Trusted Vendors, User Privileges, and Browser Control nodes in each device rule node.

For more information, see [Rule Items](#).

Device Rule Validation

Type	Rule
Host Name or IP Address	Use this device client rule to apply Allowed Items, Denied Items, Trusted Vendors, and User Privileges rules to a third party device when a user attempts to access their endpoint from a specific Host Name or IP Address. If the Host Name or IP Address is matched to the third party device, Application Control rules specific to the device are applied.
Computer Group Membership	<p>Use this device client rule to apply Allowed, Denied, Trusted Vendors, and User Privileges rules to a third party device that is a member of a specific security group. Application Control checks to see if the computer is a member of the specified security group before applying the rules.</p> <hr/> <p> If entering the Computer Group Membership details manually, you must use the fully qualified name. For example, CN=ComputerGroup, OU=Department, OU=Corporation, DC=CoreDomain.</p> <hr/>
OU Membership	Use this device client rule to apply Allowed, Denied, Trusted Vendors and User Privileges rules to a third party device that is a member of a specified Organizational Unit (OU).

Active Directory (AD) based client conditions convert the NetBIOS name of the client, obtained from Windows Terminal Server (or Citrix equivalent), to a FQDN used to query AD. The FQDN cannot be resolved if the terminal server is in the parent domain and is trying to resolve the FQDN of a connecting device in a child domain. This impacts Device and Custom rules, with Active Directory based client conditions, that are applied to terminal servers and VDIs in a root domain.

The terminal server must be configured with the DNS suffix of all child domains. The search list must be configured on all terminal servers wanting to resolve names for connecting in child domains.

For example, for the parent domain.local, the child domains, childa.domain.local and childb.domain.local, must be configured on the terminal server in order for AD based conditions to evaluate correctly.

For information about configuring domain suffix search lists, see: <https://support.microsoft.com/en-gb/kb/275553>

Custom Rules

Custom rules apply settings to devices hosting the Application Control Agent and configuration. You can add items to the Allowed Items, Denied Items, Trusted Vendors, User Privileges, and Browser Control nodes in each group rule node.

For more information, see [Rule Items](#).

Custom rules allow security control rules to be applied when certain conditions are met. You can specify conditions for the following:

- Computer
- Directory membership
- Environment
- Files and folders
- Registry
- Session and client
- The user

You can also create custom scripted conditions using Visual Basic or Java Script.

For more information on conditions for custom rules, see [Condition Management](#).

For example, you can create a custom rule that allows only users who belong to the Finance OU and who are not working on laptops to self-elevate to install a specific accounting application.

If you select the Custom Rules node, the All Custom Rules summary displays the Rule Name and the Security Level.

To add a custom rule, click the **Add Rule** drop-down arrow on the Rules ribbon and select **Custom Rule**.

To remove a custom rule, select a rule and click **Remove Rule** on the Rules ribbon. A confirmation message displays. Click **Yes** to confirm the removal.

Support for Custom Rules from Earlier Versions

Custom rules in version 10.0 differ considerably from Custom rules in version 8.8 and 8.9. You can upgrade version 8.8 and 8.9 configurations that contain Custom rules by opening them in a version 10.0 console and saving them. This recreates the Custom rules by using the new version 10.0 conditions, matching the behavior of the earlier version rules.

If you do not upgrade a version 8.8 and 8.9 configuration, the Application Control Agent version 10.0 still reads the configuration, but the URL Redirection and Custom rules are ignored. The rest of the configuration still applies.

Scripted Rules

Scripted rules allow custom rules to be created using Windows PowerShell or VB Scripts. The success or failure of the Script determines whether the security level, Allowed Items, and Denied Items that are part of the rule apply to the user.

Scripted rules can take advantage of any interface accessible via PowerShell or VBScript, such as COM (Component Object Model) and

Each script is evaluated under the following circumstances:

- When a new configuration is deployed to the computer.
- When a user logs on.

You create and edit scripts in the Scripted Rule dialog, which you access as follows:

1. In the Rules ribbon, select **Add Rule**.
2. In the drop down menu, select **Scripted Rule**.

The Scripted Rule work area displays.

You can define when the script is to be run using the following Scripted Rule Options:

- **Run script once per logon session as the logged on user** - The script runs for each user logging on. Settings are only applied for the duration of the user session.
- **Run script once per logon session as the SYSTEM user** - The script runs with SYSTEM account permissions once for each user logging on. Settings are only applied for the duration of the user session.
- **Run script once per computer as the SYSTEM user** - The script runs with SYSTEM account permission once at computer startup. Settings are applied to all user sessions until the computer restarts, the Application Control agent restarts or there is a configuration change.



Caution: Running scripts as the SYSTEM user can cause serious damage to your computer and should only be enabled by experienced script authors.

- **Do not execute script until user logon is complete** - Select to prevent the script from running until user logon is complete.
- **Wait for <n> seconds before script timeout** - Allows you to specify the number of seconds to allow a script to continue running before the script times out. A setting of zero (0) seconds prevents the script timeout. If a timeout occurs the result is fail and settings cannot be applied.

VBScripts

Each script is run within a hosted script engine allowing greater control over the script execution whilst providing a high degree of input and output control.

- No VBS file is used.
- No separate process is spawned.

A script must be written as a function and can contain many functions, but a main start function must be specified. The start function is run by the Application Control agent and can be used to call other functions.

The AMScriptRule COM object is built into the scripting engine and provides access to the following methods:

- `strUsername = AMScriptRule.UserName`
- `strUserdomain = AMScriptRule.UserDomain`
- `strSessionid = AMScriptRule.SessionID`
- `strStationname = AMScriptRule.WinStation`



The Microsoft standard in this instance means that WinStation returns the value of the name of the Terminal Services Session, which is determined by the type of session with typical values being 'Console' or 'RDP-Tcp#34', instead of the Window Station name which is typically WinSta0.

The AMScriptRule COM object also includes the following methods:

- `strLog = AMScriptRule.Log "My Log Statement"`

Allows you to output logging strings to the agent log file for use with debugging scripted rules.

- `strEnvironmentvar = AMScriptRule.ExpandEnvironment ("%MyEnvironmentVariables%")`

Expands environment variables of the user running the script.



Using WScript. shell to expand environment variables only returns SYSTEM variables.

Windows PowerShell Scripts

If the script returns (exits) with a value of 0, the script will pass and the rules are applied. If any non-zero value is returned, the script will fail and the rules will not apply.

Each PowerShell script is executed in an instance of PowerShell.exe and as such Application Control neither enforces nor adds any specific syntax – all correctly formed PowerShell will work.



PowerShell must be installed on any endpoints that will be using the script.

Add a Scripted Rule

1. Click the **Add Rule** drop-down arrow on the Rules ribbon and select **Scripted Rule**.

A new rule is added to the All Scripted Rules work area. The **Scripted Rule** dialog displays.

2. To enter a script, do one of the following:
 - Type the script in the Current Script area.
 - Open an existing script in a script editor and copy/cut the content and paste.
3. Select **Click here to edit the script**. Click **Import** to import an existing script.

Edit a Scripted Rule

1. Use the Scripted Rule dialog to create and maintain rules based on custom VB and PowerShell Scripts that are run whenever a user logs on.
2. To open the Scripted Rule dialog for a specific rule, you can either:
 - Navigate to the scripted rule in the navigation pane and select it.
 - Select the **Rules** node in the navigation tree. In the All Rules dialog, double-click the rule that you want to edit.

The Scripted Rule dialog displays.

3. Click **Click here to edit the script**.

The Configure this Scripted Rule dialog displays.

4. In the Script tab, add or amend the script to be used when your users log on.
5. In the Options tab, select the script execution setting from the list of available options in the Define the execution settings section.
6. To specify the script time settings, select the appropriate options in the Define the script time settings section.
7. Click **OK**.

Sample scripts

Scriptable rule to determine if an AAC filter has been passed Using VBScript

The following VBscript demonstrates how to control the applications to which a user has access.

```
Function ScriptedRule()  
'Name of Filter scan expected to pass  
ExpectedFilter = "FWALL"  
'Get Server Name  
Set objNTInfo = CreateObject ("WinNTSystemInfo")  
ServerName = lcase (objNTInfo.ComputerName)  
'Set initial return value  
ScriptedRule = False  
'Create MetaFrame Session Object  
Set MFSession = Createobject ("MetaFrameCOM.MetaFrameSession")  
'Initialize the session filters for this session  
For Each x in MFSession.SmartAccessFilters  
'return true if our filter is found  
If x = ExpectedFilter Then  
ScriptedRule=True  
AMScriptRule.Log "SmartAccessFilter match found."  
End If  
Next  
End Function
```

Scriptable rule to determine if a computer is in a Computer OU Using VBScript

The following VBscript can be used to determine if a computer is in a Computer Organizational Unit:

```

Function ScriptedRule()
ScriptedRule = vbFalse
strCompName = AMScriptRule.StationName
Set oRootDSE = GetObject("LDAP://RootDSE")
strDNSDomain = oRootDSE.Get("DefaultNamingContext")
Set oOU = GetObject("LDAP://OU=TheOUyouAreSearching,OU=Parent,OU=Parent," & strDNSDomain)
oOU.GetInfo
For each member in oOU
    If UCase(strCompName) = UCase(member.CN) Then
        ScriptedRule = vbTrue
        Exit For
    End If
Next
End Function

```

Scriptable rule to determine if a user is a member of a certain OU Using VBScript

The following sample VBScript shows the main components of a script and demonstrates how to access information about the username of the user logging on to the system, and match with a specific domain and organizational unit:

```

Function MyScript()
'Get the username of the user logging in (also works when running as SYSTEM)
strUserName = AMScriptRule.UserName
'Get the domain of the user logging in (also works when running as SYSTEM)
strUserDomain = AMScriptRule.UserDomain
'Look up user environment variables (when running as SYSTEM, only SYSTEM variables are available)
strClientName = AMScriptRule.ExpandEnvironment ("%ClientName%")
'Log the output
AMScriptRule.Log strUserName & " logged in on " & strClientName
'Check if the user is a member of the domain
If strUserDomain = "MyDomain" Then
    'If so, see if the user is in the MyOU OU
    Set objOU = GetObject ("LDAP://ou=MyOU,dc=MyDomain,dc=com")
    objOU.Filter = Array("user")
    For Each objUser In objOU
        'Check if there is a match with the user logging on
        If objUser.sAMAccountName = strUserName Then
            'if there is, then set the function to True
            MyScript = True
        End If
    Next
End If
'Unless there is a username match, the function defaults to False
End Function

```

Scriptable rule to determine if a user is a member of a certain OU Using Windows PowerShell

The following sample Windows PowerShell script shows the main components of a script and demonstrates how to access information about the username of the user logging on to the system, and match with a specific domain and organizational unit:

```

#Script checks if the current user is a member of the OU specified
# Return 0 if TRUE
# 1 otherwise
$loginuser = $env:username
$bindpt = [adsis] "LDAP://OU=TS_Users,OU=Users,OU=MyUser,OU=MyOU,DC=MyDomain,DC=com"
$users = New-Object System.DirectoryServices.DirectorySearcher $bindpt
$users.Filter = "(&(objectClass=User)(sAMAccountName=$loginuser))"

```

```
$obj = $users.FindOne()
if($obj -eq $null)
{
#" Not a Member"
exit 1
}
```

Process Rules

The Process node allow security control rules to be matched with specific requesting processes. Process rules allow you to manage access for an application to run child processes which might otherwise be managed differently in other rules. You can add Allowed Items, Denied Items, Trusted Vendors and User Privilege Management to the rule.

For further information, see [Rule Items](#).

You can add files, folders, drives, signature items, network connection items and application groups as managed items into the Allowed Items and Denied Items lists of a process rule.

The Process Rule only manages the first level of child process run by the application, not the children of child processes. The Process does not manage the application. This must be managed by other rules unless the application is managed as a child process in another Process Rule.

Create a Process Rule

The process rule applies to the application that is attempting to start an application, load a component, or access a network resource. The process rule can allow certain applications to run but prohibit it from running when launched by specific processes.

- Rules are displayed in the order they are created and are not alphabetical.
- Process rule names must be unique. You cannot create two process rules with the same name.
- You cannot have duplicate processes.
- You cannot cut, copy and paste process rules.

1. From the Rules ribbon, select the **Add Rule > Process Rule**.

A process rule is created and consists of four rule items: Allowed Items, Denied Items, Trusted Vendors, and User Privileges.

2. Right-click the new process rule and select **Rename**.
3. Give the rule an intuitive name.
4. Apply the required security level: **Restricted**, **Audit Only** or **Unrestricted**.

For more information, see [Security Levels](#).

5. Add a process to the rule.
6. Add an item to a rule item.

Add a Process to a Process Rule

Use the Process Rule work area to add processes to a process rule. The processes listed within this area are used during rules processing to match the rule to a request's process originator.

The first column displays the name and location of the process file or signatures, the second contains the signature for the process, if applicable, and the third column displays the description of the process, if present.

1. Select the process rule.

The Process Rule work area is displayed.

2. In the Rules ribbon, select the **Add Process** drop-down arrow and do one of the following:
 - To add a file, select **Add > File**
 - To add a signature, select **Add > Signature**

You can add multiple files at once. You can drag and drop files from Windows Explorer or another file manager, and cut, copy, and paste.



You cannot have duplicate processes.

3. Use the Rescan Signature button on the Rules ribbon to

Add an Allowed or Denied Item to a Process Rule

Allowed Items and Denied Items can contain files, folders, drives, signatures, Windows Store Apps, and network connection items. They can also include groups.

1. Select the required Allowed or Denied Item.
2. Click the **Add Item** drop-down arrow on the **Rule Items** ribbon and select either **Allowed** or **Denied**.
3. Once you have selected the item type to add to the process rule, do one or more of the following:
 - To add a file, select **Add > File**
 - To add a folder, select **Add > Folder**
 - To add a drive, select **Add > Drive**
 - To add a signature item, select **Add > Signature Item**
 - To Add a Network Connection item, select **Add > Network Connection Item**
 - To Add a Windows Store App, select **Add > Windows Store App**
 - To add a group, select **Add > Group**
4. Groups can consist of a number of items. For example, all the File, Folder, Drive, and Signature File items for a particular application.

Example: Using a Process Rule to Restrict Access to FTP

You can use process rules to allow, for example, only certain applications to access FTP.

This example shows how to use process rules to allow only a specific application to access FTP ports 20 and 21. The first step is to create a group to specify the

Step 1 - Create a Group

1. Select the Group Management node.
2. Select **Add Group** on the Groups ribbon.
3. Select and right-click the new group and select **Rename**.
4. Rename the group with an intuitive name, for example, *Specify FTP Ports*.
5. Select the **Add Item** drop-down arrow on the Groups ribbon and select **Network Connection**.

The Add a Network Connection dialog displays.

6. Specify the host in the Host field.

7. Select the **Ports** button on the right hand-side of the Ports field. The Common Ports dialog displays.
8. Select ports **20** and **21: FTP - Data Port** and **FTP - Control port**, and click **Add**.
9. Select the **Text contains wildcard characters** option and click **Add**.

Step 2 - Create a Process Rule to Block Access to FTP Ports 20 and 21

1. Select the top level Process rule node.
2. Select the **Add Rule** drop-down arrow on the Rules ribbon and select **Process Rule**.
3. Select and right-click the new process rule and select **Rename**.
4. Give the rule an intuitive name, for example, *Cannot access FTP*.
5. Right-click within the Processes work area, and select **Add > File**.
The Add a File dialog displays.
6. Enter * in the File field and click **Add**. This denotes that all files are blocked from accessing ports FTP 20 and 21. The use of
7. Expand the new process rule node.
8. Select the Denied Items node.
9. Select the **Add Item** drop-down arrow and select **Denied > Group**. The **Group selection for** dialog box displays.
10. Select the group created in the Create a Group procedure and click **Add**. This rule now prohibits all applications from accessing the FTP ports 20 and 21.

Step 3 - Create a Process Rule to Allow Access to FTP Ports 20 and 21

1. Select the top level Process rule node.
2. Select the **Add Rule** drop-down arrow on the Rules ribbon and select **Process Rule**.
3. Select and right-click the new process rule and select **Rename**.
4. Give the rule an intuitive name, for example, *Can access FTP*.
5. In the Processes work area, right-click and select **Add > File**.
The Add a File dialog displays.
6. Browse to and select the file that you want to access FTP, for example, Internet Explorer.
7. If required, expand the new process rule node.
8. Select the **Allowed Items** node.
9. Select the **Add Item** drop-down arrow and select **Allowed > Group**. The **Group selection for** dialog displayed.
10. Select the group created in the Create a Group procedure and click **OK**. This rule now allows the specified application to access the FTP ports 20 and 21.

Step 4 - Set the Group Rule to Restricted

1. Expand the Group node and select **BUILTIN\Administrators**. The **Group Rule** work area displays.
2. Drag the Security Level slider to **Restricted**.

Step 5 - Save the configuration

Save the configuration. Only the application specified in the procedure can access FTP ports 20 and 21. All other applications cannot.

Rule Options

[Allowed Items](#), [Denied Items](#), [Trusted Vendors](#), and [User Privileges](#) can be applied to Files, Folders, Drives, Digital Signatures, Signature Items, Network Connection Items, Windows Store Apps, and Group Items.

See [Rule Items](#) for more details on adding items.

Each rule option must be associated with one or more [Rule Type](#).

Rule Matching

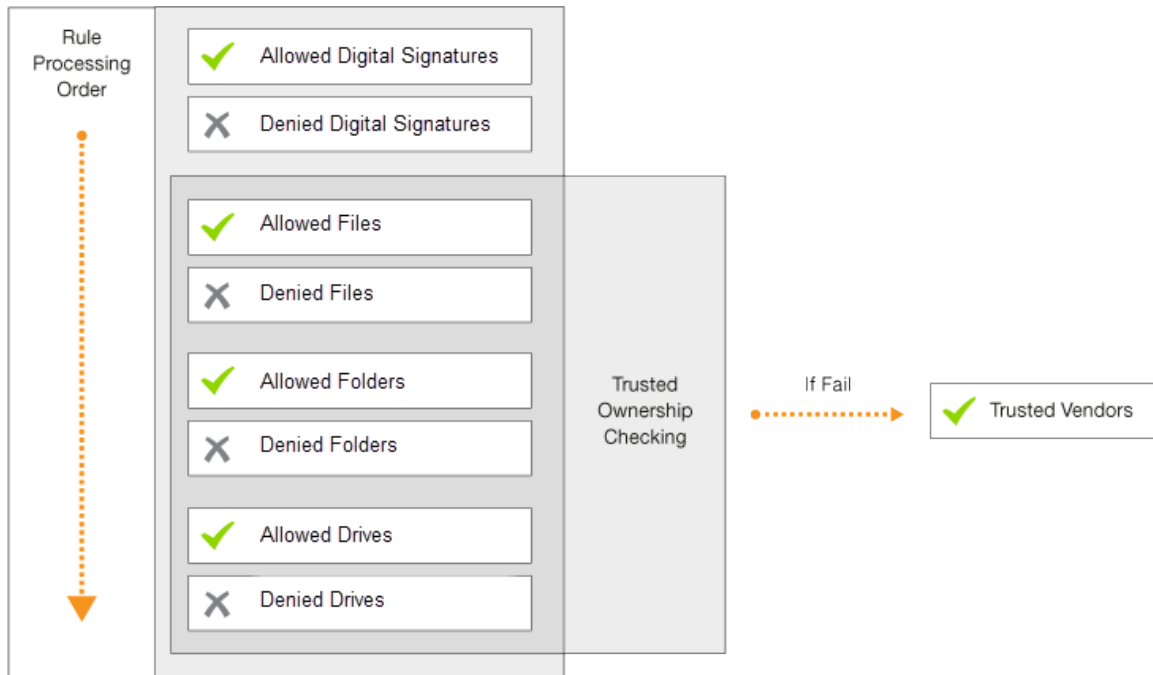
Rule matching takes place when Application Control intercepts a file execution request and checks the configuration policy to determine whether a file is allowed to run.

Applying Rule Policies

The most lenient security policy is applied to a user profile that is affected by more than one rule. For example, a user who matches both a user rule assigned the Restricted security level and also a group rule that assigns the Self-Authorizing security level is granted self-authorizing privileges for all decisions and application use.

Matching Files and Rules

The Application Control agent applies rules by making a suitable match for the file type.



Matching is based on a three stage approach that considers security, matching order, and policy decisions:

1. Security:
 - Is the user restricted?
 - Is ownership of the executable item trusted?
 - Where is the executable located?
2. Matching:
 - Does the executable match a signature?
 - Does the executable match an allowed or denied Item?
3. Policy:
 - Is Trusted Ownership checking enabled?
 - Is there a timed exception?
 - Is there an application limit?

Trusted Ownership Checking

During the rule matching process, Trusted Ownership checking is performed on files, folders, and drives to ensure that ownership of the items is matched with the list of trusted owners in the default rule configuration. If the check fails, a Trusted Vendor check is initiated.

If configured, when an executed file matches the predefined list of Allowed items, an additional security check ensures the ownership matches a user in the trusted owners list. If this check fails, Application Control attempts to match the digital signature of the file with the Trusted Vendor list. If a match still cannot be found the execution will be blocked.



Trusted Ownership checking is not necessary for items with digital signatures as these cannot be imitated.

When a default configuration is used, any new or existing file introduced to the system that is overwritten or renamed has its ownership changed to the current user. As a result, if the change is made by an untrusted user, any future execution requests will fail the trusted owners check.

Trusted Ownership checking can be used as a global rule or on a per item basis. To stop Application Control checking for trusted ownership, ensure **Enable Trusted Ownership Checking** is not selected in **Global Settings > Trusted Owners**.

Apply and Remove Rules

The Group rules node allows you to match security control rules with specific user groups within the enterprise.

The Group summary displays the group name, textual Security Identifier (SID) and security level of the rule. An SID is a data structure of variable length that identifies user, group, and computer accounts. Every account on a network is issued a unique SID when the account is first created. Internal processes in Windows refer to an accounts SID rather than the accounts user or group name. Likewise, Application Control also refers to a user or group SID unless the SID could not be found when added to the configuration.

Apply Rules

1. From the Rules Items ribbon, click **Add**.
2. Select the type of rule to be created:
 - Group
 - User
 - Device
 - Custom
 - Scripted
 - Process

The Add Rule dialog displays.

3. Enter the relevant information and click **OK**.

Remove Rules

1. To remove a group rule, highlight a rule and click **Remove Rule** on the Rules ribbon.

A confirmation message displays
2. Click **Yes** to confirm the removal.

Metadata

Add a File

Properties Metadata Access Times Application Limits

This rule item will only apply to files that have the following selected metadata.

General

Product Name ☒ Windows® Internet Explorer
☐ Use regular expression

Vendor ☒ Microsoft Corporation
☐ Use regular expression
☒ Verify certificate at runtime Verify Options

Company Name ☒ Microsoft Corporation
☐ Use regular expression

File Description ☒ Internet Explorer
☐ Use regular expression

File Version

Minimum ☒ 10.0.9200.21684 Maximum ☒ 10.0.9200.21684

Product Version

Minimum ☒ 10.0.9200.21684 Maximum ☒ 10.0.9200.21684

Populate metadata from file

Add Cancel

Metadata adds additional criteria for matching files and folders, once a match has been made with the file or folder properties. For example, adding metadata for a vendor, allows you to verify that a file is signed by a particular verified publisher.

Metadata is available for files and folders in allowed, denied, and user privilege application rule items. Metadata can be entered manually or added from an existing file. Select the **Metadata** tab for a file or folder for a compatible rule item:

- To add metadata from a file, select the Metadata tab and click **Populate metadata from file** and select the file from which you want to use the metadata. Select the check boxes for the required metadata.
- To add metadata manually, select a check box and add the required data.

To view the metadata for a file using Windows Explorer, right-click the file and select **Properties**. Metadata is displayed in the Details tab.

The following metadata can be configured for file and folder items:

General

- **Product Name** - The name of the product.
- **Vendor** - If the file has been digitally signed, the vendor name associated with the signature. A further option is available to test that the vendor metadata of the file can be trusted.

If Vendor metadata is enabled, a further option becomes available - **Verify certificate at runtime**. When this option is enabled, the agent verifies the certificate whilst it is matching the file. Click **Verify Options** to access a further set of criteria, used during file matching.

For further information, see [Verify Options](#).

- **Company Name** - The name of the company that produced the product.
- **File Description** - The file or folder description as defined by the vendor or company.

The information displayed can be amended to criteria, which can include segments of the metadata, wildcards (*) can be used.

File Version

- **Minimum** - Displays the minimum version number for the selected file.
- **Maximum** - Displays the maximum version number for the selected file.

The information displayed can be amended to introduce a version range, where the maximum and minimum version number can be defined using wildcards and all versions of the file that falls between the range can be monitored.

Product Version

- **Minimum** - Displays the minimum product version number for the selected file.
- **Maximum** - Displays the maximum product version number for the selected file.

The information displayed can be amended to introduce a version range, where the maximum version number can be defined using wildcards and all versions of the versions of the product that falls between the range can be monitored.

Wildcards can be used to substitute parts of the metadata information to allow you specify a required match based a segment of the selected metadata. For Example, if you had a vendor of Microsoft Corporation, but wanted anything associated with Microsoft, you could replace the word "Corporation" with a wildcard (*) to match anything associated with Microsoft not specifically "Microsoft Corporation".

Rule items will only apply to files that match the selected metadata.

Allowed Items

Add Allowed items to group rules to grant users access to specific items without providing them with full administrative privileges. The Allowed items are displayed in the Allowed Items list under a selected group rule:

Files

If a filename alone is specified, for example, myapp.exe, then all instances of this are allowed regardless of the location of the application. If the file is specified with the full path, for example, \\servername\sharename\myapp.exe, then only this instance of the application is allowed. Other instances of this application need to satisfy other Application Manager rules to be granted execution. For the files and folders in Application Manager that refer to items on a DFS share you need to specify the target server, rather than the Namespace server in the UNC path.

For more information, see [Distributed File Systems](#).

Folder

A complete folder may be specified, for example, \\servername\servershare\myfolder, and all applications within this folder, and all subfolders if required, allowed to execute. No checks are made on the files within the folder and as such any file copied into this folder will be allowed to execute. Select **Include subdirectories** to include all directories beneath the specified directory. If you add a network file or folder path you must use the UNC name, as the Application Manager agent ignores any paths that are configured where the Drive letter is not a local fixed disk. The user can access the network application through a network mapped drive letter, as the path is converted to UNC format before validating it against the configuration settings. To automatically apply environment variables, select **Substitute environment variables where possible** in the **Add a file** or **Add a folder** dialogs. This makes the paths more generic for applying on different machines. Wildcards support provides an additional level of control for specifying generic file paths.

Drive

You can specify a complete drive, for example, W, and all the applications on this drive are allowed to execute, including subfolders. No checks are made on the files in the drive so any file copied into any folder on this drive is allowed to execute.

Signature Item

A file may be added along with a digital hash of the file. This ensures that only that particular file may be executed but from any location. For more information, see Signature Hashing.

Network Connection Item

A Network Connection Item can be specified. All files on the network are allowed to execute.

Windows Store

Choose which Windows Store apps are allowed. You can select one of the following:

- Allow **All Installed Apps**
- Allow the selected **Individual Apps**
- Allow all apps by a named **Publisher**

Groups

Groups can contain any number and combination of items, for example, the File, Folder, Drive, Signature, and Network for a particular application. All files are allowed to execute.

Add an Allowed Item

1. Select the Allowed Items node in **Rules > Group > Everyone**.
2. Click **Add Item** and from the drop-down arrow select **Allowed**.
3. Select the item that you want to make allowed, for example File.

The Add a File dialog displays.

4. Enter or browse for the file to be made allowed.

The **Substitute environment variables where possible** checkbox is selected by default. If it is not selected, environment variables will not be replaced with a generic environment variable.

5. If applicable, enter any further information relating to the allowed item, in the Description field.
6. Check **Allow file to run even if it is not owned by a trusted owner** if you want the file to run regardless of the owner.

The selected application is listed in the Allowed Items work area.

Remove an Allowed Item

1. Select the Allowed Items node in **Rules > Group > Everyone**.
2. Highlight the item to be removed.
3. Click **Remove Item** in the Rule Items ribbon.

The Remove Items dialog displays.

4. Click **Yes** to remove the item or **No** to abort the task.

The selected application is listed in the Allowed Items work area.

Access Times

Access times allow you to specify what time and on what days a particular application is allowed to be run and can be applied to Allowed Items in Groups, Users, Devices, Custom Scripts, and Process Rules. Access periods can only be assigned when you check the **Only allow files to run at certain access times** option in the Access Times tab when adding or amending an allowed Item. Times can be amended using the Access Times option from the Rule Items ribbon. Access times can be added for file, folder and signature allowed items.

Assign Access Times

Add a File

Properties Metadata **Access Times** Application Limits

☒ Only allow files to run at certain access times

Right-click on a day to create a new access period. You can adjust these periods once created. Access is denied by default.

Remove All Periods

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
6 AM	All Day	All Day	All Day	All Day	All Day	All Day	All Day
7:00							
8:00							
9:00							
10:00							
11:00							
12 PM							
1:00							
2:00							
3:00							
4:00							
5:00							
6:00							
7 PM							

Add Cancel

This task explains how to assign access times to an allowed item:

1. Select the **Allowed Items** node in **Rules > Group > Everyone**.

For the purpose of this example, the Everyone group is being used. This will vary depending on the group you select.

2. Click **Add Item** and from the drop-down arrow select **Allowed**.

3. Select the item that you want to make allowed, for example File.

The Add a File dialog displays.

4. Enter or browse for the file to be made allowed.
5. From the Access Times tab, select **Only allow files to run at certain access times**.
6. Right-click on the time and day an item can be accessed and select **New Allowed Period**. Repeat this step above to add any other access times.
7. When the allowable periods have been selected, click **Add**.

Application Limits

Application Limits allow you to specify how many times an application can be run by a user during a session. You can configure limits when you check the **Enable application limits** option located in the Application Limits tab when you add or edit an Allowed item. You can use the Application Limits option from the Rule Items ribbon once you have added an item to a rule. Session-based Application limits can only be applied to Allowed Items in the Group, User, Device, Custom, Scripted, and Process rules. You can configure a message to displays to the user when the time limit is exceeded by using the Message Settings dialog, which you can access from the Global Settings ribbon.

Apply Application Limits

The screenshot shows a window titled "Add a File" with four tabs: Properties, Metadata, Access Times, and Application Limits. The "Application Limits" tab is selected. Inside the tab, there is a section titled "Limit the number of application instances each user can have". Below this title, there is a checkbox labeled "Enable application limits" which is checked. Underneath the checkbox, it says "Limit to a **maximum** of" followed by a spinner box containing the number "3", and then "application instances per user". A note below this says "For example, setting a value of 3 will **prevent** the fourth instance of this application from being launched." At the bottom right of the dialog, there are two buttons: "Add" and "Cancel".

1. Select the Allowed Items node in **Rules > Group > Everyone**.

For the purpose of this example, the Everyone group is being used.

2. Click **Add Item** and from the drop-down arrow select **Allowed**.
3. Select the item that you want to make allowed, for example, File.
The Add a File dialog displays.
4. Enter or browse for the file to be made allowed.
5. From the Application Limits tab, select **Enable application limits**.
6. Select the application limit.
7. Click **Add**.

Allowed Items and Trusted Ownership

By default, trusted ownership checking is enabled, therefore an application must always pass trusted ownership checking if it is enabled, even if the application is an allowed item. Although trusted ownership checking can be disabled completely, this is not recommended. However, if you need to provide a user with access to file, folders or groups that are not owned by a trusted user then you can disable the trusted ownership check when creating or editing the item by checking the **Allow File to run even if it is not owned by a trusted owner** option.

Denied Items

Denied Item nodes are sub-nodes automatically created in any Rule node when you create a new rule. They allow you to add items to which the groups, users and devices specified in the rule are refused access.

If you are using the default option, which trusts all locally installed Trusted Owner applications, you only need to add specific applications that you do not want users to run. For instance, you can add administrative tools, such as management and registry editing tools.

You do not need to use this list to deny applications that are not owned by an administrator because they are blocked by trusted ownership checking.

Application Control drag and drop functionality can be used to add files, folders, drives and signature items from Windows Explorer or copy or move items between the Allowed Items node and Denied Items nodes in each of the main configuration nodes.

You can add the following items:

Files

If a filename alone is specified, for example, myapp.exe, then all instances of this are denied regardless of the location of the application. If the file is specified with the full path, for example, \\servername\sharename\myapp.exe, then only this instance of the application is denied. Other

instances of this application need to satisfy other Application Control rules to be granted execution. For the files and folders in Application Control that refer to items on a DFS share you need to specify the target server, rather than the Namespace server in the UNC path.

For more information, see [Distributed File Systems](#).

Folder

A complete folder may be specified, for example, \\servername\servershare\myfolder, and all applications within this folder, and all subfolders are denied. No checks are made on the files within the folder and as such any file copied into this folder will be denied. Select **Include subdirectories** to include all directories beneath the specified directory. If you add a network file or folder path you must use the UNC name, as the Application Control agent ignores any paths that are configured where the Drive letter is not a local fixed disk. The user can access the network application through a network mapped drive letter, as the path is converted to UNC format before validating it against the configuration settings. To automatically apply environment variables, select **Substitute environment variables where possible** in the **Add a file** or **Add a folder** dialogs. This makes the paths more generic for applying on different machines. Wildcards support provides an additional level of control for specifying generic file paths.

Drive

You can specify a complete drive, for example, W, and all the applications on this drive, including subfolders, are denied. No checks are made on the files in the drive so any file copied into any folder on this drive is denied.

Signature Item

A file may be added along with a digital hash of the file. This ensures that only that particular file may be executed but from any location. For more information, see Signature Hashing.

Network Connection Item

A Network Connection Item can be specified. All files on the network are denied.

Windows Store

Choose which Windows Store apps are denied. You can select one of the following:

- Allow **All Installed Apps**
- Allow the selected **Individual Apps**
- Allow all apps by a named **Publisher**

Groups

Groups can contain any number and combination of items, for example, the File, Folder, Drive, Signature, and Network for a particular application. All files are denied.

Add a Denied Item

To add an item, select the Denied Items node and click the **Add Item** drop-down arrow on the Rule Items ribbon, select **Denied** and select the type of Denied Item you want to add.

This task prevents all users accessing an application on a network share:

1. Select the Denied Items node in **Rules > Group > Everyone**.
2. Click **Add Item** in the Rule Items ribbon and select **Denied**.
3. Select **File**.

The Add a File dialog displays

4. Enter or browse for an application, for example, regedit.exe.

The selected application is listed in the Denied Items work area.

5. Attempt to run the selected application.

The application is denied and a message box displays with the notification that the application is not authorized.

Remove a Denied Item

1. Select the item to remove in the Denied Items node.
2. In the Rule Items ribbon, click **Remove Item**.
3. Click **Yes** in the confirmation dialog.

The item is removed from the node.

Trusted Vendors

Trusted Vendors can be specified in each Application Control rule node. Trusted Vendors are used for listing valid digital certificates. A digital certificate is an electronic document that uses a digital signature to bind together a public key with an identity. This includes information such as the name of a person or organization, address, and so on. Digital certificates are issued by a certificate authority and used to verify that a public key belongs to an individual. Application Control queries each file execution to detect the presence of a digital certificate. If the file has a valid digital certificate and the signer matches an entry in the Trusted Vendor list, the file is allowed to run, and overrides any Trusted Ownership checking.

You can check whether a file has a digital certificate by displaying the Properties dialog. A file has a digital certificate if there is a Digital Signatures tab in which you can view details of the certificate including, signer information, advanced settings and an option to display the certificate.

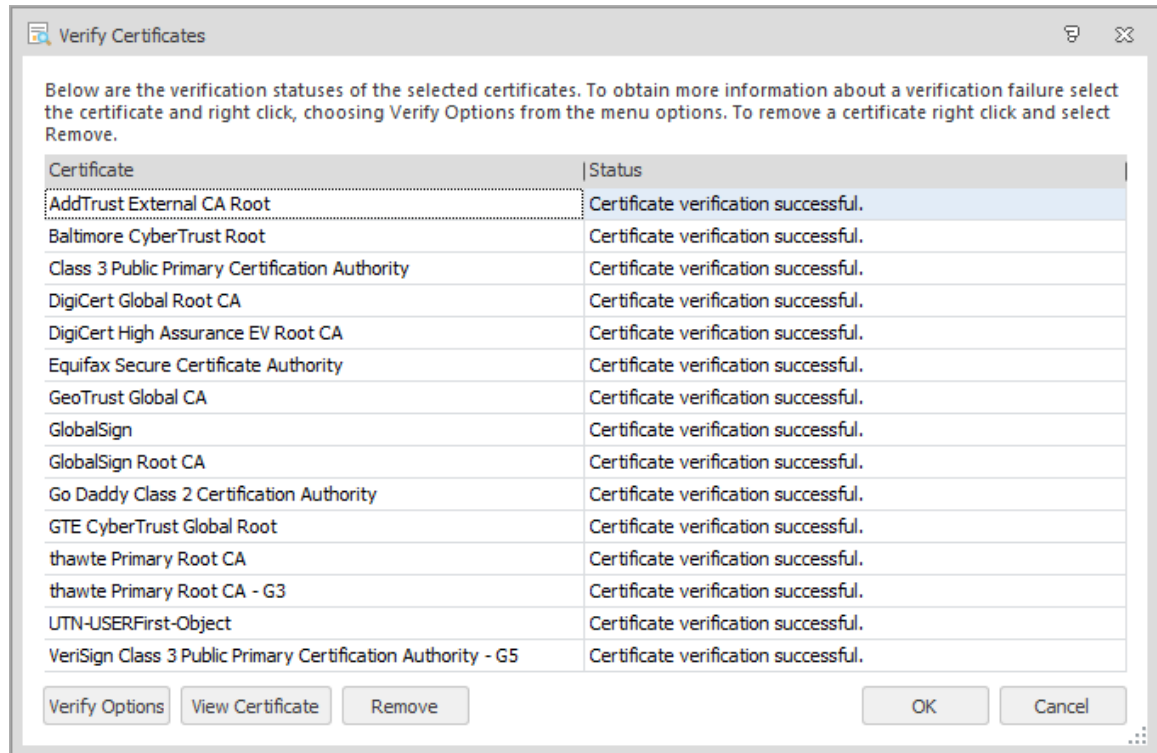
The Trusted Vendors sub node is available in each rule node, for listing valid digital certificates.

Add a Certificate to a Trusted Vendor

1. Select the **Trusted Vendors** node to which you want to add the certificate.
2. Click the **Add** drop-down arrow on the Rule Items ribbon and select the required option:
 - **From Signed File** - Select a known file that has already been signed by the vendor whom you want to trust. Application Control can then identify the vendor's specific signature to identify additional code from that same vendor.
 - **Import File-based Store** - Add certificates from a P7B file, created in a file-based store, such as Certificate Manager.

3. Navigate to the required file and click **Open**.

The Verify Certificates dialog lists the name(s) of all added certificates. The Status column shows whether the certificate has been validated successfully or if any errors have been detected.



Further options are available for the listed certificates. Highlight the required certificate and select the one of the following:

- **Verify Options** - Check the status of the certificate, enforce the certificate's expiry date and apply advanced certificate options.

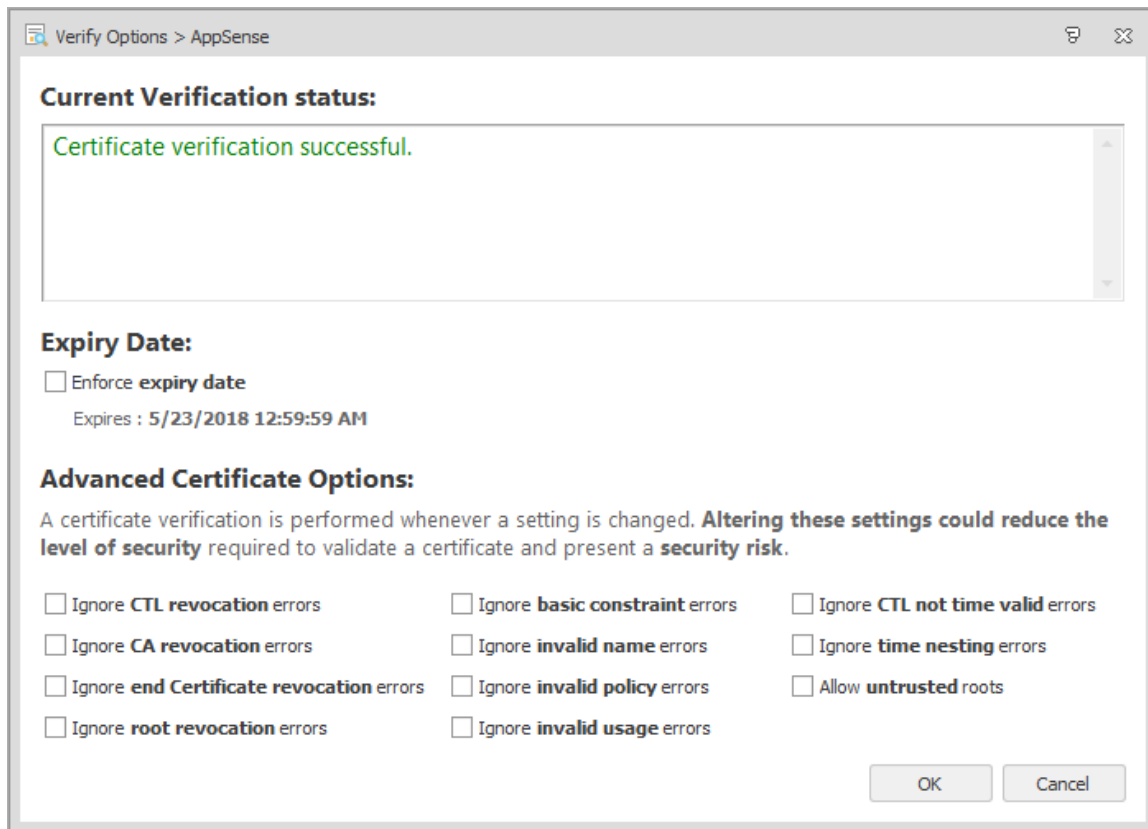
For further information, see [Verify Options](#).

- **View Certificate** - View more information about the selected certificate.
- **Remove** - Remove the selected certificates and prevent them from being added to the trusted vendor. Multiple certificates can be select and removed using the Shift and Ctrl keys.

4. Click **OK**.

The listed certificates are added to the Trusted Vendors work area.

Verify Options



Verify options for Trusted Vendors allow you to specify parameters for validating a certificate by ignoring or allowing specific attributes. The certificate must be valid for the rule to be applicable, but there are different levels of validation with which you can configure a certificate.

The advanced options are available when adding metadata for files, by clicking **Verify Options**.

Altering the settings using the Advanced Certificate Options could reduce the level of security required to validate a certificate.

The Verify Options dialog displays the current status of a certificate and gives access to Expiry Date and Advanced Certificate options. The verify options are available from:

- Certificates for trusted vendors
- Metadata for allowed or denied files and folders

When you add a certificate, Application Control checks to see if it is valid and displays the result of the check in the Current Verification Status message box. The check is performed each time an option in this dialog is updated. For example, the certificate could be invalid due to an untrusted root certificate. If the *Allow untrusted roots* option is subsequently selected, Application Control checks the certificate again and updates the status to show that certificate validation is successful.

You can also choose whether to enforce the expiry date of the certificate. The default setting is that Application Control ignores the expiry date of certificates so they remain valid indefinitely. If you choose to enforce the expiry date, the certificate is unverified after that date and the vendor is no longer trusted.

Advanced Certificate Options

Advanced certificate options allow you to specify parameters for validating a certificate by ignoring or allowing specific attributes. The certificate must be valid for the rule to be applicable, but there are different levels of validation with which you can configure a certificate.

Altering the settings using the Advanced Certificate Options could reduce the level of security required to validate a certificate and present a security risk.

Apply the following settings when determining certificate verification:

- **Ignore CTL revocation errors** - Ignore errors when obtaining Certificate Trust List (CTL) revocation.
- **Ignore CA revocation errors** - Ignore errors when obtaining Certificate Authority (CA) revocation.
- **Ignore end Certificate revocation errors** - Ignore errors when obtaining the end certificate, or user certificate, revocation is unknown.
- **Ignore root revocation errors** - Ignore errors when obtaining valid root revocation.
- **Ignore CTL not time valid error** - Ignores that the certificate trust list is not valid, for example, the certificate may have expired.
- **Ignore time nesting errors** - Ignores that the Certificate Authority (CA) certificate and the issued certificate have validity periods that are not nested.



The CA certificate may be valid from January 1st to December 1st, and the issued certificate from January 2nd to December 2nd. This means that the validity periods are not nested.

- **Ignore basic constraint errors** - Ignores that the basic constraints are not valid.
- **Ignore invalid name errors** - Ignores that the certificate has an invalid name.
- **Ignore invalid policy errors** - Ignores that the certificate has an invalid policy.
- **Ignore invalid usage errors** - Ignores that the certificate was not issued for the current use.
- **Allow untrusted roots** - Ignores that the root cannot be verified due to an unknown certificate authority.

User Privilege Rules

In the User Privileges node for any rule, you can select the User Privilege Policies to be applied to files, folders, signatures, groups, and Windows Components when the rule is matched. You can configure self-elevation to allow a user to run an item with elevated user privileges. You can also use system controls to control the uninstallation or modification of selected applications, the management of specified services, and the clearing of event logs.

Select the User Privileges node for a rule and the work area includes four tabs - Applications, Components, Self-Elevation and System Controls.

Applications

Click **Add Item** in the Privilege Management ribbon to add a file, folder, signature, or group to the Applications tab. The item is listed in the tab under the columns Item, Policy, and Description. To change the policy applied to the file, folder, or signature, double-click the item to access the edit dialog box. Select the policy to apply from the **Policy** drop-down list.

For more information on adding items, see [Rule Items](#).

Components

Because Management Console snap-ins and Control Panel Applets are not executables, they cannot be elevated using a single executable but instead must be elevated using command line matching. The User Privileges Management (UPM) components section provides easy shortcuts to configuring these items that are equivalent to an Add File UPM policy with specified arguments.



Command line arguments and spawning mechanisms will vary depending on the Operating system your individual users are using.

Control Panel components and Network Adapter features and functions are typically controlled by explorer.exe. Elevating explorer.exe to run in the context of a Local Administrator is not ideal as this can open up a range of security issues. To resolve this and enable the user to access the functionality under the context of an administrator without opening the entire explorer shell, User Privileges Management places the AppSense Control Panel components in the Windows Control Panel alongside existing components. These can now be controlled at an access level specific to the function, without changing any rights associated with explorer.exe.



Use the filter in the Select Components dialog to filter the supported components by operating system.

For more information, see the [Components](#) table for a list of components that are specific to particular operating system.

Example: Applying a User Rights Policy to a Control Panel Component

1. Expand the applicable Group rule in the navigation pane and select the **User Privileges** node.
2. Select the **Components** tab in the work area.
3. In the Privileges Management ribbon, select **Add Item > Add Component**.

The Select Components dialog displays.

4. Select the components you want the user to run as an administrator, for example, Add and Remove Programs\Programs and Features.
5. Click **OK**.

The component is now listed in the Components tab.

6. Do one of the following:
 - To elevate the privileges for the selected component, select **Builtin Elevate** from the drop-down in the User Rights Policy column.
 - To restrict the privileges for the selected component, select **Builtin Restrict** from the drop-down in the User Rights Policy column.
7. Save the configuration.

Self-Elevation

Self-Elevation can be applied to signatures, files and folders items that would usually require administrative privileges to run and function. Self-Elevation provides an option from the Windows Explorer context menu to run an item with elevated rights. When a user attempts to elevate a specified item, a prompt can be configured to request that the user enters a reason for the elevation before it is applied.

For more information, see [Self-Elevation](#).

System Controls

System Controls are used to allow or prevent named services being stopped, event logs being cleared and specific applications being uninstalled or modified.

For more information, see [System Controls](#).

Browser Control

In the Browser Control node, you can:

- Configure URL redirection
- Add elevated websites
- Add web installations
- Import snippets



When a new configuration containing Browser Control items, such as URL Redirection, is deployed to endpoints, users need to close and re-open browsers before the configuration can take effect. Closing and re-opening the browsers enables the browser extensions. If an existing configuration with Browser Control is updated with additional Browser Control items, the updated configuration takes effect as soon as it is deployed. The browser extensions are already enabled, so it's not necessary to close and reopen browsers.

Configure URL Redirection

Use this feature to automatically redirect users when they attempt to access a sensitive URL. By defining a list of denied or sensitive URLs, you redirect any user attempting to access a listed URL to a default warning page or a custom page.

Before you configure this feature for Internet Explorer, you must enable third-party browser extensions using Internet Options for each of your endpoints. Alternatively, this can be applied via Group Policy.



URL Redirection is compatible with Internet Explorer 8, 9, 10, and 11. When using Chrome, all managed endpoints must be part of a domain.

URL Redirection is configured in the Add URL to Redirect dialog accessed from the Browser Control ribbon. The URL Redirection functionality is enabled or disabled for the application in Advanced Settings, accessible via the Manage ribbon.



In versions prior to Application Control 10.0, URL Redirection was a global setting accessed via the Manage ribbon. Configurations containing URL Redirections that were created in versions 8.8 and 8.9 of the product can be opened in the console and automatically upgraded in version 10.0. The URL Redirections are converted to Custom rules that contain the following:

- Matching conditions for connection types, IP addresses, and port numbers.
- Browser Control items for the sensitive URLs (listed on the URL Redirection tab).

If you don't upgrade the configuration, the version 10.0 agent still reads the configuration, but the URL Redirection and Custom rules are ignored. The rest of the configuration still applies.

Enable or Disable URL Redirection

1. In the Manage ribbon, click **Advanced Settings**.
The Advanced Settings dialog displays.
2. In the Policy Settings tab, go to the Functionality section and either select or deselect the **Enable URL Redirection** checkbox.
3. Click **OK**.

Add URL Redirection to a Rule

1. In the Application Control navigation pane, select the Browser Control node for the rule to which you want to add URL redirection.
2. In the Browser Control ribbon, select **Add Item > Add URL**.
The Add URL to Redirect dialog displays.
3. Enter the denied URL.
4. Choose the response when a user attempts to access the prohibited URL:
 - **Display the default warning page when a URL is redirected** - the user is directed to the default "Access is denied" page.
 - **Display a custom page when a URL is redirected** -specify an alternative location instead of displaying the default warning page. For example, this could be a location within your organizations network, a file on a disk, your intranet or another website.
5. Enter an optional description for your future reference.
6. Click **Add**.

The prohibited URL, the redirection URL, and any description you added are now listed in the columns on the URL Redirection tab of the Browser Control work area.

An AAMP configuration is created with the Elevated Website feature configured and can be deployed to your endpoints. When a User accesses the specified webpage, the original browser will be redirected to a warning page and a new instance of IE is spawned. The new browser will have full administrative rights and permit any components to be run.

Add a Web Installation

A number of Web Installations require the end user to have administrative rights. For example, an ActiveX control such as Adobe Flash Player or a web download such as Microsoft Silverlight.

The Web Installation feature of Browser Control allows the elevation to administrative privileges for ActiveX installers from a particular domain. You can create a basic configuration whereby you enter the name of the domain only, or you can create an advanced configuration and specify the CAB file for an item, its Class ID, and the minimum and maximum versions. You can also specify that only signed controls from the domain can be installed.

1. Navigate to the **Browser Control** node under your selected rule.
2. In the Browser Control ribbon, select **Add Item > Add Web Installation**.
The Add New Web Installation dialog displays.
3. Enter a descriptive Name for the web installation.
4. If you want to allow only **signed controls**, select the relevant checkbox.
5. Enter the **Website URL** for the installation. For example, enter *adobe.com* to allow installations from all of adobe.com.
6. Click **Add**.

The Websites tab in the Browser Control work area displays the name of the new web installation.

Add a Web Installation (Advanced Settings)

1. Navigate to the **Browser Control** node under your selected rule.
2. In the Browser Control ribbon, select **Add Item > Add Web Installation**.
The Add New Web Installation dialog displays.
3. Enter a descriptive Name for the web installation.
4. If you want to allow only **signed controls**, select the relevant checkbox.
5. Select **Use advanced settings**.
The Advanced Settings section becomes active.
6. Enter the Installer URL, for example *http://www.example.com/control.cab*.
7. Complete the following fields, or leave them blank to be ignored: Class ID, Minimum Version, and Maximum Version
8. Click **Add**.

The Websites tab in the Browser Control work area displays the name of the new web installation.

Example: Create a Configuration that Allows the Installation of Adobe Flash Player

A common scenario is a standard user attempting to download and install Adobe Flash Player. This requires administrative privileges. When an attempt is made, the User Account Control (UAC) dialog is displayed requesting the user to enter an administrative password. Many organizations do not want to give the users administrative privileges.

1. Select the **Browser Control** node under your selected rule.
2. In the Browser Control Ribbon, select **Add Item > Add Web Installation**.
The Add New Web Installation dialog displays.
3. Enter a name for the Web Installation in the Name field, for example, *Adobe Flash*.

4. Enter the URL in the Website URL field. For example, adobe.com, to allow installations from all of adobe.com.
5. Ensure the **Only allow signed controls option** is selected.
6. Click **Add**.

The Websites tab in the Browser Control work area displays the name of the new web installation.

7. Ensure the default **Builtin Elevate** policy is selected in the Policy column of the Websites tab.
8. Save the configuration. All downloads that are signed and are from the specified website are allowed.

Along with the above procedure other configurable items need to be considered. For example, for an ActiveX installation you would need to allow the ActiveX file to run, and any executables that the control calls. You need to consider Process rules, Trusted Vendors, any Digital Certificates, Allowed Items, Elevated items, and so on.

Import Snippets

Snippets give Application Control the ability to import and merge partial configurations into a currently open configuration in the console.

This is particularly useful for web installations because, along with creating the web installation part of the configuration, a number of other configurable items need to be considered. These include Process Rules, Allowed Items, Trusted Vendors, any Digital Certificates, Elevated items, and so on.

The latest snippets can be downloaded by logging onto the Support Portal.

Download Recent Snippets from MyAppSense

1. Select a rule.
2. In the Browser Control ribbon, select **Import Snippet**.
The Import Snippet dialog displays.
3. Click the **Support Portal** link in the dialog.
The most recent snippets are displayed.
4. Select a snippet and save it to C:\Program Files\AppSense\Application Control\Console\Snippets. This is the default location.
The snippet is now available in the Import Snippet dialog.
5. Select the snippet and click **Add**.

6. To view what is included in the snippet click the **View the items that will be added to the configuration** link.

A configuration report displays.

7. Click **Continue**.

The snippet is imported and you can view the items in the various nodes in the console.

Add Elevated Websites



This feature is only supported in 32-bit versions of Internet Explorer 8, 9, 10 and 11.

The Elevated Website feature allows you to define a particular URL which opens in a separate secured, but elevated, instance of Internet Explorer. When elevated, the user is granted administrative privileges allowing them to install and execute components such as additional software or ActiveX controls specific to the site.

Before you configure this feature, you must enable third-party browser extensions using Internet Options for each of your endpoints, alternatively this can be applied via Group Policy.

It is recommended that this feature is only used for internal websites which require elevation to run content such as diagnostic tools or a moderated portal containing administrator approved software.



You should not elevate websites that may allow users to obtain software which may pose a security risk to your network; such as pop-ups, search bars or external links.

1. Select the **Browser Control** node under your selected group.
2. Select the Browser Control ribbon.
3. Click **Add Item** and select **Add Elevated Website**.

The Add New Elevated Website dialog displays.

4. Enter a meaningful description for your reference.
5. Enter the web address in the Website URL field.

You can use regular expressions to define websites. To use this functionality, select **Use regular expression** and enter the website URL criteria. For example, `https://.+\.com$` elevates and redirects any secure websites with the .com extension - such as `https://www.cisco.com`, but does not elevate and redirect `http://www.cisco.com`

For more information, see [Wildcards and Regular Expressions](#).

6. Click **Add**.
7. Save the AAMP file.

Rules Items

Rule items include files, folders, network drives and connections, signature files, Windows Store Apps, and groups, which you can add to rule nodes, such as Allowed Items and User Privileges.

Files

Add Files to Allowed or Denied Item for a Rule

1. In the navigation pane, select the Allowed Item or Denied Item node for a rule.
2. In the Rule Items ribbon, select either:
 - **Add Item > Allowed > File.**
 - **Add Item > Denied > File.**

The Add a File dialog displays.

3. In the Properties tab, click the ellipsis (...) in the text box, navigate to the file that you want to add and click **OK**.
4. If required, you can select the following:
 - Substitute environment variables where possible
 - Use regular expressions
5. Enter optional command line arguments in the **Arguments** text box. Enter all arguments as they appear in Process Explorer.

Command line arguments extend the matching criteria beyond what is entered in the File field. If an argument is added, both file and argument must be satisfied for a match to occur. Any argument that appears on the command line for a process, such as flags, switches, files, and guids, can be added.

For examples of valid argument, see [Arguments Example](#).

6. To add metadata to the file, select the **Metadata** tab:
 1. Click **Populate metadata from file**.
 2. The following fields can be populated: Product Name, Vendor, Company Name, File Description, File Version, and Product Version.
 3. Select the checkboxes for the metadata to refine criteria for the file.

If Vendor metadata is enabled, a further option becomes available - **Verify certificate at runtime**. When this option is enabled, the agent verifies the certificate whilst it is matching the file. Click **Verify Options** to access a further set of criteria, used during file matching.

For further information, see [Verify Options](#).

7. To specify that the file may run at specific access times only, select the **Access Times** tab:
 1. Select **Only allow files to run at certain access times**.
 2. To specify a specific allowed period, right-click the time period in the calendar area, and select **New Allowed Period**.
8. To limit the number of instances of an application a user can have, select the **Application Limits** tab:
 1. Select **Enable Application Limits**.
 2. Enter the limit in the spinbox.
9. Click **Add** to add the file to the Allowed/Denied Items for the rule.

Add a File for User Privilege Management for a Rule

1. In the navigation pane, select the **User Privileges** node for a rule.
2. In the User Privileges ribbon, select **Add Item > Application > File**.

The Add a File for User Privilege Management dialog displays.
3. In the Properties tab, click the ellipsis (...) in the text box:
 1. In the Open dialog, navigate to the file that you want to add and click **OK**.
 2. If required, you can select the following:
 - Substitute environment variables where possible
 - Use regular expressions
 - Make file an Allowed Item

4. Enter optional command line arguments in the **Arguments** text box. Enter all arguments as they appear in Process Explorer.

Command line arguments extend the matching criteria beyond what is entered in the File field. If an argument is added, both file and argument must be satisfied for a match to occur. Any argument that appears on the command line for a process, such as flags, switches, files, and guids, can be added.

For examples of valid argument, see [Arguments Examples](#).

5. To apply a policy, select the policy from the drop-down in the Policy section.

You can select the following options for the policy:

- Apply to child processes
 - Apply to common dialogs
 - Install as a trusted owner
6. If required, enter an optional description of the file for your future reference.

7. To add metadata to the file, select the **Metadata** tab:

1. Click **Populate metadata from file**.
2. The following fields can be populated: Product Name, Vendor, Company Name, File Description, File Version, and Product Version.
3. Select the checkboxes for the metadata to refine criteria for the file.

If Vendor metadata is enabled, a further option becomes available - **Verify certificate at runtime**. When this option is enabled, the agent verifies the certificate whilst it is matching the file. Click **Verify Options** to access a further set of criteria, used during file matching.

For further information, see [Verify Options](#).

8. Click **Add** to add the file to the User Privilege Management for the rule.

Arguments Example

Denied File	Allowed File	Result
shutdown.exe	shutdown.exe Arguments: -r -t 30	shutdown.exe runs only when -r -t 30 is on the command line - anything else run by shutdown.exe is denied.

To configure the arguments of an allowed or denied item correctly, they must appear as they do in Process Explorer for example:

File: C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE

Command line: "C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE" /n C:\example.docx

Would be configured as:

File: Absolute or relative path of winword.exe

Arguments: /n C:\example.docx

Folders

Add a Folder to Allowed or Denied Items List for a Rule

In the navigation pane, select the Allowed Item or Denied Item node for a rule.

1. In the Rule Items ribbon, select either:
 - **Add Item > Allowed > Folder.**
 - **Add Item > Denied > Folder.**

The Add a Folder dialog displays.

2. In the Properties tab, click the ellipsis (...) in the text box, navigate to the folder that you want to add and click **OK**.
3. If required, select the following:
 - Substitute environment variables where possible
 - Use regular expressions
 - Include subfolders
4. To add metadata to the folder, select the **Metadata** tab:

1. Click **Populate metadata from file**.
2. The following fields can be populated: Product Name, Vendor, Company Name, File Description, File Version, and Product Version.
3. Select the checkboxes for the metadata to refine criteria for the file.

If Vendor metadata is enabled, a further option becomes available - **Verify certificate at runtime**. When this option is enabled, the agent verifies the certificate whilst it is matching the file. Click **Verify Options** to access a further set of criteria, used during file matching.

For further information, see [Verify Options](#).

5. Click **Add** to add the folder to the Allowed/Denied Items for the rule.

Add a Folder for User Privilege Management for a Rule

1. In the navigation pane, select the **User Privileges** node for a rule.
2. In the User Privileges ribbon, select **Add Item > Application > Folder**.

The Add a Folder for User Privilege Management dialog displays.

3. In the Properties tab, click the ellipsis (...) in the text box:
 1. In the Open dialog, navigate to the file that you want to add and click **OK**.
 2. If required, you can select the following:
 - Substitute environment variables where possible
 - Include subfolders
 - Use regular expressions
 - Make folder an Allowed Item
4. To apply a policy, select the policy from the drop-down in the Policy section.

You can select the following options for the policy:

- Apply to child processes
 - Apply to common dialogs
 - Install as a trusted owner
5. If required, enter an optional description of the folder for your future reference.
 6. To add metadata to the file, select the **Metadata** tab:
 1. Click **Populate metadata from file**.
 2. The following fields can be populated: Product Name, Vendor, Company Name, File Description, File Version, and Product Version.
 3. Select the checkboxes for the metadata to refine criteria for the folder.

If Vendor metadata is enabled, a further option becomes available - **Verify certificate at runtime**. When this option is enabled, the agent verifies the certificate whilst it is matching the file. Click **Verify Options** to access a further set of criteria, used during file matching.

For further information, see [Verify Options](#).

7. Click **Add** to add the file to the User Privilege Management for the rule.

Drives

Add a Drive to the Allowed or Denied Items Lists for a Rule

1. Select the Allowed Items or Denied Items node for a rule.
2. In the Rule Items ribbon, select either:
 - **Add Item > Allowed > Drive**.
 - **Add Item > Denied > Drive**.
3. The Add a Drive dialog displays.
4. Enter the drive letter and an options description for your future reference.
5. Click **Add** to add the drive to the list of allowed or denied items for the rule.

Signatures and Signature Items

Add a Signature to the Allowed or Denied Items List for a Rule.

1. In the navigation pane, select the Allowed Item or Denied Item node for a rule.
2. In the Rule Items ribbon, select either:
 - **Add Item > Allowed > Signature Item.**
 - **Add Item > Denied > Signature Item.**

The Add a Signature dialog displays.

3. In the Properties tab, click the ellipsis (...) in the text box. In the Open dialog, navigate to the file, for example an EXE file, that you want to add and click **OK**.

The Signature Hash Value field is populated with the signature hash value of the file.

4. To specify that the file may run at specific access times only, select the **Access Times** tab:
 1. Select **Only allow files to run at certain access times.**
 2. To specify a specific allowed period, right-click the time period in the calendar area, and select **New Allowed Period.**
5. Click **Add** to add the signature file to the Allowed/Denied Items for the rule.

Add a Signature File to User Privilege Management for a Rule

1. In the navigation pane, select the **User Privileges** node for a rule.
2. In the User Privileges ribbon, select **Add Item > Application > Signature.**

The Add a Folder for User Privilege Management dialog displays.

3. In the Properties tab, click the ellipsis (...) in the text box:
4. In the Open dialog, navigate to the file that you want to add and click **OK**.
5. If required, you can select Make signature file an Allowed Item
6. Enter optional command line arguments in the **Arguments** text box.
7. To apply a policy, select the policy from the drop-down in the Policy section.

You can select the following options for the policy:

- Apply to child processes
 - Apply to common dialogs
 - Install as a trusted owner
8. If required, enter an optional description of the folder for your future reference.
 9. Click **Add** to add the signature file to the User Privilege Management for the rule.

Network Connection Items

Network Connection Items can be created for any network resource and can be added directly to a Rule. Adding single Network Connection Items to Allowed and Denied Item lists is useful when a more granular level of control is required, or when only a few items are required. However, using this method could prove time-consuming.

Network Connection Items can be cut, copied or dragged and dropped between rules. There are no default Network Connection Items in a configuration. The full path of the Network Connection Item cannot exceed 400 characters.

Add a Network Connection Item to the Allowed or Denied Item List for a Rule

1. In the navigation pane, select the Allowed Item or Denied Item node for a rule.
2. In the Rule Items ribbon, select either:
 - **Add Item > Allowed > Network Connection.**
 - **Add Item > Denied > Network Connection.**

The Add a Network Connection dialog displays.

3. Select the connection type:
 - IP Address - Select to control access to a specific IP Address.
 - Network Share - Select to control access to UNC paths. The prefix \\ is added to the Host field.
 - Host Name - Select to control access to a specific Host Name.

4. Complete the connection details. The combined number of characters for all three fields, Host, Port and Path must not exceed 400.

- **Host** - The IP Address or Host Name for the network connection. This depends on the type of connection selected. You can use the ? and * wildcards. Additionally, ranges can be used for IP Addresses, which are indicated by use of a hyphen (-). An IP Address must be in IP4 octal format, for example, n.n.n.n. If Network Share is selected as the connection type, the \\ prefix is required.

The full path for the target resource can be entered in Host, for example
http://server1.company.local:80/resource1/.

Move focus away from Host and the path is automatically split into the separate connection options:

- http:// is removed from the Host field and server1.company.local remains.
- : is removed and 80 is moved to Port.
- /resource1/ is moved to Path.
- **Port** - The port number of the network connection. This can be used in combination with IP Address or Host Name to control access to a specific port. Ranges and comma separated values are allowed as a part of the port number. Click **Ports** to display a list of commonly used ports. Select as many ports as required.
- **Path** - The path of the network connection. You can use the ? and * wildcards. To use wildcards in the path, select the **Text contains wildcard characters** option.



The path is only relevant for controlling HTTP and

- **Description** - Enter a meaningful description to describe the network connection.
5. Click **Add** to add the network connection to the list of Allowed or Denied Items for the rule.

Windows Store Apps

Add a Windows Store App to the Allowed or Denied Item List for a Rule

1. In the navigation pane, select the Allowed Items or Denied Items node for a rule.
2. In the Rule Items ribbon, select either:
 - **Add Item > Allowed > Windows Store App.**
 - **Add Item > Denied > Windows Store App.**
3. Select the required option:
 - **All Installed Apps** - Include any app that users have installed.
 - **Individual Apps** - Include specific apps selected from built-in snippets and snippets downloaded from appsense.com. Use the Version Matching drop-down to target the required app versions.
 - **Publisher** - Include all apps from a named publisher. You can enter publisher details manually or extract details from a locally installed app.
4. Click **OK**.

Groups

Groups can be added to User Privileges to hold and manage logical collections of files, folders, drives, signature files, and network connection items. You can also add them to the lists of Allowed or Denied Items for a rule.

Add a Group to the Allowed or Denied Items List for a Rule

1. In the navigation pane, select the Allowed Item or Denied Item node for a rule.
2. In the Rule Items ribbon, select either:
 - **Add Item > Allowed > Group.**
 - **Add Item > Denied > Group.**

The Group Selection dialog displays listing the available groups.

3. Select the groups you want to add.
4. If you want to execute all the listed rule items regardless of the owner, select the **Allow Untrusted Owner** checkbox for the app.
5. Click **OK**.

Add a Group to User Privilege Management for a Rule

1. In the navigation pane, select the User Privileges node for a rule.
2. In the User Privileges ribbon, select **Add Item > Applications > Group**.

The Group Selection dialog displays. The available groups are listed.

3. To assign the User Privileges rules to the selected group, select **Add To Rule**.
4. You can also select the following options:
 - Policy - Select the policy - for example Builtin Elevate - from the drop-down list.
 - Make Allowed - Make the selected group allowed and overwrite any associated allowed items.
 - Allow Untrusted owner - Execute all the listed rule items regardless of the owner. This option becomes available when Make Accessible is selected.
 - Apply to Child Processes - Apply the policy to all the children and other descendants of the parent process.
 - Apply to Common Dialogs - Allow the open and save dialogs to run with administrative privileges when selected from an elevated process.
 - Install as Trusted Owner - Make local administrators the owner of all files created by the defined application.
5. Click **OK**.

Control Applications

You can combine Application Control's security methods - such as Trusted Ownership Checking - with rules in a configuration to control which users can install and run applications.

Application Control uses a method known as Trusted Ownership checking to prevent the execution of any user-introduced executable. Only applications installed by Trusted Owner - for example, administrators - are allowed to run by default. In the case of Microsoft applications such as Project and Visio that have been installed in a multi user environment, you can use Application Control to allow access only to these applications by specified licensed device.

The Application Control configuration contains two Group rules. These are **BuiltIn\Administrators**, who are unrestricted and can run any executable, and **Everyone**, who can only run executables owned by Trusted Owners. Each rule created has an Allowed Items and Denied Items list.

The Allowed Items list allows administrators to give access to executables that would normally be blocked by default rules, for example Trusted Ownership failure or Network Executables.

The Denied Items list allows administrators to deny access to executables that would normally be allowed by default rules.

Because Microsoft applications will often be licensed to run on only a few devices, it is best practice to use Application Control to initially deny access to the application for everyone, then allow access to the few, based on the allowed device.

Step 1 Restrict Access to an Application for Everyone

1. Expand the **Group > Everyone** node.
2. Right-click the **Denied Items** node and select **Add Item > Denied > File**. The Add a File dialog displays.
3. Browse to and select the application to restrict access to, or enter the name in the File field, and click **Add**. All standard users are now denied from using the specified application.

The above configuration denies access to everyone, therefore you must create an exception rule to allow named licensed devices to run the application. The devices can be specified using an IP address range or NetBIOS name. These devices are the connecting client machine in a terminal server/Citrix environment.



Application Control rules operate differently to Microsoft Group Policies in that an Allowed Item rule overrides any Denied Item rule.

Step 2 Create an Exception Rule

1. In the Rules ribbon, select **Add Rule > Device Rule**.

A new rule is created.

2. Right-click the new rule and select **Rename**.
3. Type an intuitive name such as *Visio Licensed Devices*.
4. Expand the new rule.
5. Select the **Allowed Items** node.
6. In the Rule Items ribbon, select **Add Item > Allowed > File**.

The Add a File dialog displays.

7. Browse to and select the application to make allowed to authorized devices, or enter the name in the File field, and click **Add**.

This is the same application that you have restricted in Step 1.

Step 3 Specify Authorized Devices

1. Select the new Device rule.
2. Select **Add Client Device** on the Rules ribbon.

The Add a Client Device dialog displays

3. Browse to and select the devices to authorize for the specified application and click **Add**.

You can also specify the devices by directly typing:

- IP Address (for example, 192.168.1.80)
- IP Address Range (for example, 192.168.1.10-20)
- NetBIOS name (for example, Ivanti-PC1)

You can include any combination of the above.

4. To specify that the devices are the connecting devices and not the physical devices that are running the application, select **Connecting Device** in the Device Type column for each device.

Step 4 Save the Configuration

Save the Configuration. When the configuration is deployed to a Citrix/Terminal Server only the specified devices are allowed to launch the Microsoft 'per device' licensed application

Use Process Rules to Restrict Access to FTP

You can use process rules to allow, for example, only certain applications to access FTP.

This task shows how to use process rules to allow only a particular application to access FTP ports 20 and 21. The first step is to create a group to specify the

Step 1 Create a Group

1. Select the Group Management node.
2. Select **Add Group** on the Groups ribbon.
3. Select and right-click the new group and select **Rename**.
4. Rename the group with an intuitive name, for example, *Specify FTP Ports*.
5. Select the **Add Item** drop-down arrow on the Groups ribbon and select **Network Connection**. The Add a Network Connection dialog displays.
6. Specify the host in the Host field.
7. Select the **Ports** button on the right hand-side of the Ports field. The Common Ports dialog displays.
8. Select ports **20** and **21**: **FTP - Data Port** and **FTP - Control port**, and click **Add**.
9. Select the **Text contains wildcard characters** option and click **Add**.

Step 2 Create a Process Rule to Block Access to FTP Ports 20 and 21

1. Select the top level Process rule node.
2. Select the **Add Rule** drop-down arrow on the Rules ribbon and select **Process Rule**.
3. Select and right-click the new process rule and select **Rename**.
4. Give the rule an intuitive name, for example, *Cannot access FTP*.
5. Right-click within the Processes work area, and select **Add > File**. The Add a File dialog displays.
6. Enter * in the File field and click **Add**. This denotes that all files are blocked from accessing ports FTP 20 and 21. The use of
7. Expand the new process rule node.
8. Select the Denied Items node.
9. Select the **Add Item** drop-down arrow and select **Denied > Group**. The **Group selection for** dialog box displays.
10. Select the group created in the [Create a Group](#) procedure and click **Add**. This rule now prohibits all applications from accessing the FTP ports 20 and 21.

Step 3 Create a Process Rule to Allow Access to FTP Ports 20 and 21

1. Select the top level Process rule node.

2. Select the **Add Rule** drop-down arrow on the Rules ribbon and select **Process Rule**.
3. Select and right-click the new process rule and select **Rename**.
4. Give the rule an intuitive name, for example, *Can access FTP*.
5. In the Processes work area, right-click and select **Add > File**. The Add a File dialog displays.
6. Browse to and select the file that you want to access FTP, for example, Internet Explorer.
7. If required, expand the new process rule node.
8. Select the **Allowed Items** node.
9. Select the **Add Item** drop-down arrow and select **Allowed > Group**. The **Group selection for** dialog displayed.
10. Select the group created in the [Create a Group](#) procedure and click **OK**. This rule now allows the specified application to access the FTP ports 20 and 21.

Step 4 Set the Group Rule to Restricted

1. Expand the Group node and select **BUILTIN\Administrators**. The **Group Rule** work area displays.
2. Drag the Security Level slider to **Restricted**.

Step 5 Save the configuration

Save the configuration. Only the application specified in the procedure can access FTP ports 20 and 21. All other applications cannot.

Rules Examples

Allow Access to Selected Windows Store apps

Scenario

- You are an IT Administrator using Windows 8
- You are creating an Application Control configuration
- You have created a *Corporate\CallCenter* node
- You want to grant access the certain Windows Store apps

Process

1. Expand the **Group > Corporate\CallCenter** node.
2. Select **Denied Items**.
3. Right-click in the work area and select **Add > Windows Store App**.
The Browse Windows Store Apps dialog displays.

4. Select **Apply to all Windows Store Apps on the endpoint**.
5. Click **OK** to deny access to all Windows Store apps.
6. To specify the Windows Store apps to be made allowed, select **Allowed Items**.
7. Right-click in the work area and select **Add > Windows Store App**.

The Browse Windows Store Apps dialog displays.

The dialog is populated with all the available Windows Store apps and contains three columns:

- **Display Name** - This column displays the full name of the Windows Store app.
- **Publisher** - This column displays the registered company name for any Windows Store apps.
- **Version Matching** - This column displays the version of the Windows Store app and the default matching rule of and above.

8. Select **Apply only to the Windows Store Apps selected below**.

Select this option to grant access to all Windows Store apps available on the machine being used to create the configuration file and the endpoints where the configuration is deployed.

9. Select the Windows Store apps to be allowed.

If the machine being used to create the Application Control configuration file is not compatible with Windows Store Apps, a predefined list can be imported. For more information, see [Configure Group Rules for Windows Store Apps Using Snippets](#).

10. From the Version Matching drop-down, use the version filter options to select the version criteria to be met before the selected apps can be accessed.

There are four rules options available:

- **and above** - Select this option to grant access to the current version of the application and any future versions.
- **and below** - Select this option to grant access to the application up to and including the current version only.
- **exactly** - Select this option to grant access to the current version of the application only.
- **all versions** - Select this option to grant access to all versions of the application.

When the criteria specified in the Version Matching drop-down is matched, the Allowed or Denied Item rules are then applied.

11. Click **OK**.

When the configuration is saved and deployed access to the selected applications is granted on endpoints using Windows 8 and above.

Denied Access to Selected Windows Store apps

Scenario

- You are an IT Administrator
- You are creating an Application Control configuration
- You have created a *Corporate\CallCenter* node
- You want to prohibit access to some Windows Store apps

Process

1. Expand the **Group** > **Corporate\CallCenter** node.
2. Select **Denied Items**.
3. Right-click in the work area and select **Add > Windows Store App**.

The Browse Windows Store Apps dialog displays.

The dialog is populated with all the available Windows Store apps and contains three columns:

- **Display Name** - This column displays the full name of the Windows Store app.
- **Publisher** - This column displays the registered company name for any Windows Store apps.
- **Version Matching** - This column displays the version of the Windows Store apps and the default matching rule of and above.

If multiple users have the same app installed on the machine being used to create the configuration, each version is listed and the version number detailed in the Version Matching column.

4. Select **Apply only to the Windows Store Apps selected below**.
5. Select which apps are to be explicitly denied.

If the operating system on the machine being used to create the Application Control configuration file is not compatible with Windows Store Apps, application snippets can be imported. For more information, see [Configure Group Rules for Windows Store Apps Using Snippets](#).

6. From the Version Matching drop-down, use the version filter options to select which of the selected apps are to be denied. There are four rules options available:
 - **and above** - Select this option to prohibit access to the current version of the application and any future versions.
 - **and below** - Select this option to prohibit access to the application up to and including the current version only.
 - **exactly** - Select this option to prohibit access to the current version of the application only.
 - **all versions** - Select this option to prohibit access to all versions of the application.
7. When the criteria specified in the Version Matching drop-down is matched, the Denied Item rule are then applied.
8. Click **OK**.

When the configuration is saved and deployed, users in the Corporate\CallCenter group will only have access to all Windows Store apps but denied from using a select few.

Configuring Group Rules for Windows Store apps Using Snippets

Scenario

- You are an IT Administrator
- You have created a *Corporate\CustomerServices* group rule
- You are creating a configuration on a machine that is not compatible with Windows Store Apps for users in the group rule.
- You want to use the application snippets to grant access to certain Windows Store apps for endpoints using Windows 8 and above.

Process

1. Expand the **Corporate\CustomerServices** group rule.
2. Select **Allowed Items**.
3. Right-click in the work area and select **Add > Windows Store App**.

The Browse Windows Store Apps dialog displays all the available Windows Store apps and contains three columns:

- **Display Name** - This column displays the full name of the Windows Store apps.
- **Publisher** - This column displays the registered company name for any Windows Store apps.
- **Version Matching** - This column displays the version of the Windows Store apps and the default matching rule of and above.

4. Click **Import Snippet**.

The Import Windows Store App Snippet File dialog displays.

A list of common Windows Store app snippets are included as part of the Application Control console installation and can be found in the console folder within the Application Control installed location.

Other snippets are available from www.myAppSense.com, which is opened when you click **Login to myAppSense.com Snippets page for the latest snippets**.

5. Select the application snippets to be added and click **Open**. Hold down the **Ctrl** button on your keyboard to select more than one snippet.

6. From the Version Matching drop-down, use the version filter options to select which of the selected apps are allowed.

There are four rules options available:

- **and above** - Select this option to grant access to the current version of the application and any future versions.
- **and below** - Select this option to grant access to the application up to and including the current version only.
- **exactly** - Select this option to grant access to the current version of the application only.
- **all versions** - Is applied by default to grant access to all versions of the application.

When the criteria specified in the Version Matching drop-down is matched, the Allowed or Denied Item rules are then applied.

7. Click **OK**.

When the configuration is saved and deployed, users on machines that support Windows Store apps can access the specified apps.

Condition Management

Conditions are used in Custom Rules to apply security controls based on a number of factors. You can set conditions based on the following:

- [Computer](#)
- [Scripted](#)
- [Directory membership](#)
- [Environment](#)
- [Files and folder](#)
- [Registry](#)
- [Session and client](#)
- [User](#)

You can also create custom scripted conditions using VBScript or JScript, to handle scenarios that are not supplied as standard from the Application Console.

The security controls set in the rule items such as Allowed Items and Privilege Management are applied when the criteria in the condition are true or false. You can also use regular expressions and ranges to create advanced conditions that apply to multiple matches.



In conditions that support regular expressions, you can use simple regular expressions, such as entering **[abc]** to match anything that includes any of the characters within the brackets. You can also use more complex queries, for example **^[a-f]+** matches any user name that begins with a letter from **a** to **f**.

For further information, see [Wildcards and Regular Expressions](#).

When you create a custom rule with conditions, remember that Application Control applies a timeout of 10 seconds to evaluate all the conditions for a rule. If the conditions are not evaluated within 10 seconds, the custom rule is not applied. This is especially important when creating scripted conditions, because a script that takes longer than 10 seconds to complete causes the evaluation to time out.

To view the conditions for a Custom rule, select the node for an individual Custom rule in the navigation pane. The work area displays the security level for the rule, and beneath it, a list of the conditions for the rule and whether the rule is enabled.

At the top of the list is the Conditions drop-down menu, from which you create new conditions. Alongside the menu are the following options to manage the conditions:

- Move Left
- Move Right
- Move Down
- Move Up

- Edit
- Delete


Use the Move arrows to arrange the conditions in the list. You can move conditions up and down the list or indent them to the left or right to make them children or parents of other conditions. The position of a condition in the list hierarchy determines how it is applied. Conditions at the same level are evaluated simultaneously (an OR condition). A condition that is a child of another condition evaluates only once the parent condition has been successfully executed (an AND condition). In the following example, the user either must be an administrator OR must be a member of the Finance user group AND using a laptop before the custom rule items can apply:

Condition	Enabled
<div> <div>Is Administrator is True</div> <div>OR</div> <div> <div>User Group is Equal to Ivanti\Finance</div> <div>AND</div> <div>Device is laptop</div> </div> </div>	<div>✓</div> <div>✓</div> <div>✓</div>

For any condition that queries the Active Directory, the Application Control administrator must be a member of the target domain or have sufficient permissions to access and query the domain.

If conditions are part of a live configuration, they are included in the report when you create a full report using Configuration Profiler. The report lists the relevant condition beneath each individual custom rule, using the same condition text as in the custom rule work area:

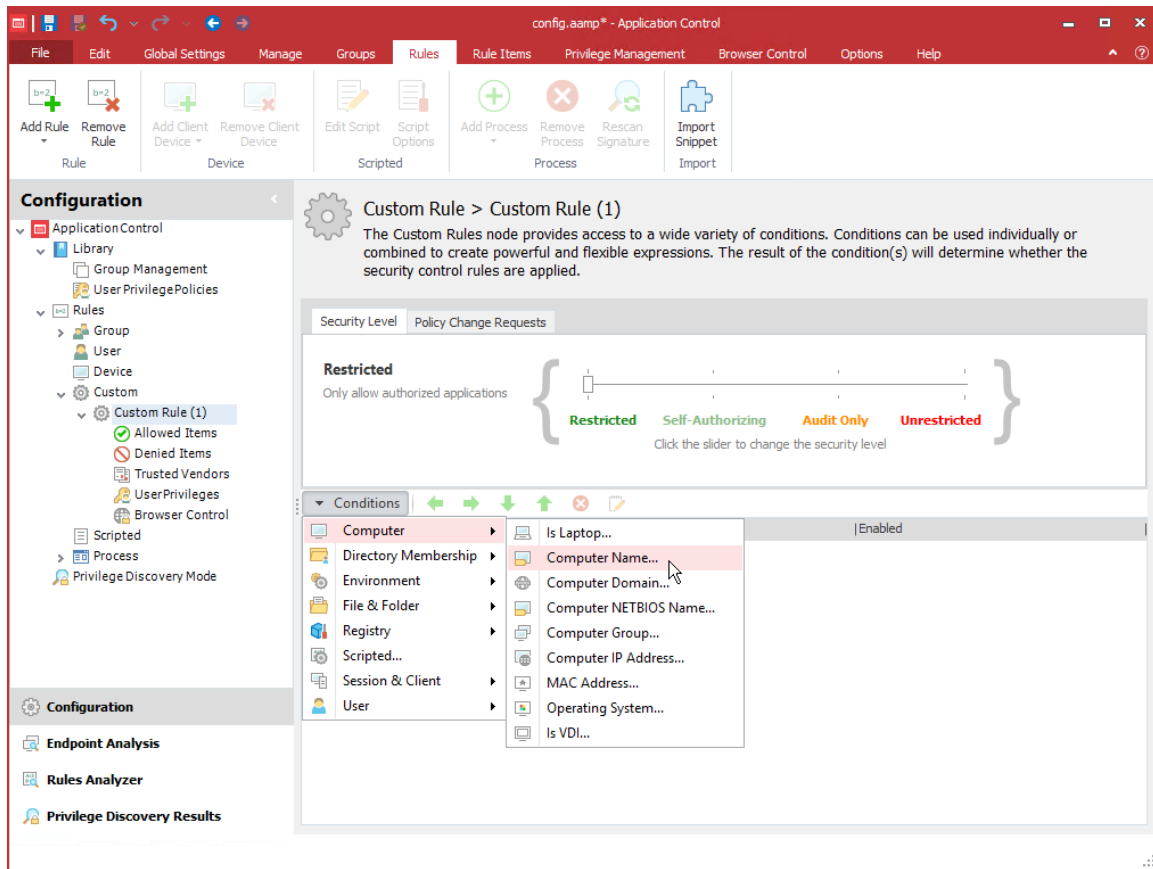
Custom Rule: Custom Rule (1)

 Security Level: Restricted

Conditions

Matching Condition
Is Administrator is True
OR User Group is Equal to Ivanti\Finance
AND Device is laptop

Create a Condition



This section applies to creating Directory Membership, User, Computer and Session & Client conditions only, as the dialog boxes these conditions use follow the same format.

1. Select the node for a Custom rule.
2. In the work area for the rule, open the Conditions drop-down menu and select the condition you want to apply, for example **Conditions > User > User Group**.

In the condition dialog, the condition tab specific to the condition type displays by default. This tab allows the parameters to be set using a common group of options and fields.

See [Condition Variables](#) for further details.

3. Define the condition using the available fields and checkboxes.
4. Select the **General** tab.
5. Enter a description and any optional notes. The description is used as the display name for conditions. If this field is left blank the display name is automatically set from the configured condition.
6. Click **OK** to save the condition.

The Application Control agent uses the condition to find a match with the same criteria for a logged on user. If a match is found, the rule and rule items attached to the condition are applied.

Active Directory Based Conditions for Devices in Child Domains

Active Directory (AD) based client conditions convert the NetBIOS name of the client, obtained from Windows Terminal Server (or Citrix equivalent), to a FQDN used to query AD. The FQDN cannot be resolved if the terminal server is in the parent domain and is trying to resolve the FQDN of a connecting device in a child domain. This impacts Device and Custom rules, with Active Directory based client conditions, that are applied to terminal servers and VDIs in a root domain.

The terminal server must be configured with the DNS suffix of all child domains. The search list must be configured on all terminal servers wanting to resolve names for connecting in child domains.

For example, for the parent domain.local, the child domains, childa.domain.local and childb.domain.local, must be configured on the terminal server in order for AD based conditions to evaluate correctly.

For information about configuring domain suffix search lists, see: <https://support.microsoft.com/en-gb/kb/275553>

Reusing Conditions

You can reuse conditions you have already created by copying and pasting them from one Custom Rule to another.

You can cut, copy, and paste whole conditions using the options in the Edit Ribbon.

Condition Variables

Each type of condition can be specified using variations of the following fields, drop-downs, and checkboxes:

- **Equal** - A comparison is made against the contents of the **Match** field to target the users or computers that fulfill those criteria. Enter the criteria into the **Match** field or use the ellipsis (...) to search or select as required.
- **Not Equal** - Targets all users or computers that do not fulfill the criteria in the **Match** field. Enter the criteria in the **Match** field or use the ellipsis to search or select as required.
- **Query** - Targets all users or computers that match the criteria specified in the **Query** field. Using wildcards in the query allows a wide range of matches, for example:
 - **Windows* - target users or computers ending in the text Windows.
 - *Windows** - target users or computers starting with the text Windows.
 - **Windows** - target users or computers containing the text Windows.
- **Regular Expression** - Use regular expressions to specify advanced queries for users or computers.

- **Between** - Used for conditions where a range of values can be set. For example, a condition can be created to apply to a selected range of IP addresses.
- **Evaluate once per session** - When selected, a condition is evaluated and the result is cached. If the condition is run again, the result is obtained from the cache rather than evaluating the condition again.

Field Validation

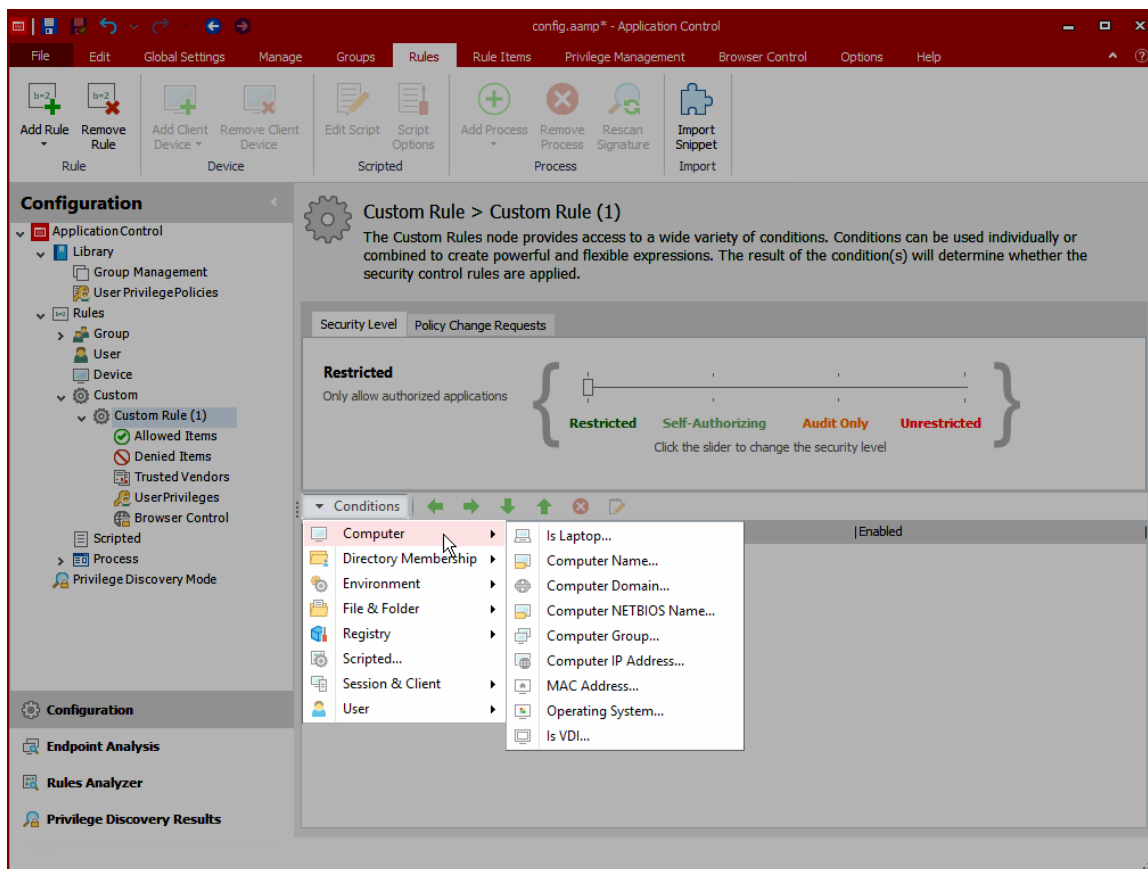
The table below lists the strings that are acceptable in the fields of the various conditions.

Condition	Field	Allowed String	Example
User Group	Match	domain\group	appsense/sales matches the group 'sales' in the appsense domain.
		LDAP	CN=sales , matches the 'sales' group in the appsense.com domain.
	Query	domain\gro*	appsense\sals* matches group names starting with 'sal' in the appsense domain.
		domain*gro	appsense*les matches group names ending with 'les' in the appsense domain.
		domain*gro*	appsense*ale* matches group names containing 'ale' in the appsense domain.
User Name	Match	domain\user	appsense\smithj matches the user name 'smithj' in the appsense domain.
	Query	domain\use*	appsense\smit* matches group names starting with 'smit', in the appsense domain.
		domain*use	appsense*ith matches group names ending with 'ith', in the appsense domain.
		domain*use*	appsense*ith* matches group names containing 'ith', in the appsense domain.
Computer Group	Match	domain\group	appsense/sales matches the group 'sales' in the appsense domain.
		LDAP	CN=sales , matches the 'sales' group in the appsense.com domain.
	Query	domain\gro*	appsense\sals* matches group names starting with 'sal' in the appsense domain.
		domain*gro	appsense*les matches group names ending with 'les' in the appsense domain.
		domain*gro*	appsense*ale* matches group names containing 'ale' in the appsense domain.
Computer	Match	computer	SalesDesk01 matches the computer name

Condition	Field	Allowed String	Example
Name			'SalesDesk01'.
	Query	comp*	SalesDesk* matches all computer names starting 'SalesDesk'.
		comp	Desk01 matches all computer names ending with 'Desk01'.
		comp	Desk* matches all computer names containing 'Desk'.
Computer Domain	Match	domain	appsense matches the domain name 'appsense'.
		domain	appsense.com matches the domain name 'appsense.com'.
	Query	dom*	app* matches all computer domains starting 'app'.
		*dom	*sense matches all computer domains ending 'sense'.
		dom	*sen* matches the domains containing 'sen'.
Computer NETBIOS	Match	computer	SalesDesk01 matches the computer NETBIOS name 'SalesDesk01'.
	Query	comp*	SalesDesk* matches all computer names starting 'SalesDesk'.
		comp	Desk01 matches all computer names ending with 'Desk01'.
		comp	Desk* matches all computer names containing 'Desk'.
Computer IP Address	Match	xxxx.xxxx.xxxx.xxxx	192.168.0.1 matches the IP address 192.168.0.1.
	Between	xxxx.xxxx.xxxx.xxxx	IP Address 1: 192.168.0.1 , IP Address 2: 192.168.0.254 matches all IP addresses between "192.168.0.1" and "192.168.0.254".
User OU Membership	Match	LDAP	CN=sales , matches the directory membership of user OU 'sales', in the appsense.com domain.
	Query	ou*	sales* matches user OU names starting with 'sales'.
		*ou	*sales matches user OU names ending with 'sales'.

Condition	Field	Allowed String	Example
		ou	*sales* matches user OU names containing 'sales'.
Computer OU Membership	Match	LDAP	CN=sales , matches the directory membership of computer OU 'sales' in the appsense.com domain.
	Query	ou*	sales* matches computer OU names starting with 'sales'.
		*ou	*sales matches computer OU names ending with 'sales'.
		ou	*sales* matches computer OU names containing 'sales'.
Directory Site	Match	sitename	testsite matches the site name 'testsite'.

Computer Conditions



These conditions target individual computers or groups of computers using various identifiers. Rules can be applied to a computer regardless of who is using it. The Application Control agent checks the specified criteria against that of the managed computer and applies any associated conditions to the computer or group of computers.



LSA support is not available on Computer conditions.

Condition	Description
Is Laptop	A condition to check whether the endpoint is a laptop. The agent checks the endpoint for a battery. If one exists, the condition returns true.
Computer Name	A condition for a specific computer. Enter the computer name directly or search using specified criteria on selected locations.
Computer Domain	A condition for a defined network of computers. Use the Name Resolution Type drop-down to specify whether the condition uses the DNS Domain or Windows Domain naming conventions. The domain entered in the Match field must be in the format used in your organization for the selected naming convention. For example, a DNS domain name is <i>testing.xyz.local</i> , whereas the Windows domain name is <i>testing</i> .
Computer NETBIOS Name	A condition for a computer identified by its NETBIOS name.
Computer Group	A condition based on a user group for a particular computer. The agent checks whether the specified active directory group or groups exist and compares the Security Identifier (SID) against the SID of the user's computer for a match. The condition only matches computers in the specified group - to include nested groups, select the Search nested groups checkbox.
Computer IP Address	A condition based on an IP address entered into the Address field. A range of IP addresses can be defined using the Between option and the two Address fields. For ranges, the IP address is not treated as a whole number but based upon the value of each octet. For example, if the range was from 190.190.190.190 to 200.200.200.200, 198.198.198.198 would pass but 198.198.210.198 would not as the third octet is not within the set range.
MAC Address	A condition defined by the Media Access Control (MAC) address of the network cards within a computer.

Condition	Description
Operating System	<p>A condition that applies only when the specified operating system is matched. The operating system can be further defined by version, service pack, build number, edition, CPU architecture and Terminal Services enabled.</p> <p>The Version text box provides a drop-down to select the operating system version. It also supports free text, allowing you to enter any RTM number. For example, if you wanted to specify Windows 8, enter the RTM number - 6.2.9200.</p> <p>For Build Number, select a condition, such as Greater than or Equal to in the drop-down, then enter a build number in the field. You cannot include a dot character(.) in the build number. If the build number is 10240.17113, for example, you enter 10240. To ensure you have the correct build number, you can check the relevant Microsoft release information. For example, to view build numbers for Windows 10 releases, go to https://technet.microsoft.com/en-us/windows/release-info</p>
Is VDI	<p>A condition which applies actions only when the endpoint is one of the following virtual desktops:</p> <ul style="list-style-type: none"> • Xen Desktop 5 • Xen Desktop 7 • VMware view • Quest vWorkspace

Scripted Rules

Scripted rules allow custom rules to be created using Windows PowerShell or VB Scripts. The success or failure of the Script determines whether the security level, Allowed Items, and Denied Items that are part of the rule apply to the user.

Scripted rules can take advantage of any interface accessible via PowerShell or VBScript, such as COM (Component Object Model) and

Each script is evaluated under the following circumstances:

- When a new configuration is deployed to the computer.
- When a user logs on.

You create and edit scripts in the Scripted Rule dialog, which you access as follows:

1. In the Rules ribbon, select **Add Rule**.
2. In the drop down menu, select **Scripted Rule**.

The Scripted Rule work area displays.

You can define when the script is to be run using the following Scripted Rule Options:

- **Run script once per logon session as the logged on user** - The script runs for each user logging on. Settings are only applied for the duration of the user session.
- **Run script once per logon session as the SYSTEM user** - The script runs with SYSTEM account permissions once for each user logging on. Settings are only applied for the duration of the user session.
- **Run script once per computer as the SYSTEM user** - The script runs with SYSTEM account permission once at computer startup. Settings are applied to all user sessions until the computer restarts, the Application Control agent restarts or there is a configuration change.



Caution: Running scripts as the SYSTEM user can cause serious damage to your computer and should only be enabled by experienced script authors.

- **Do not execute script until user logon is complete** - Select to prevent the script from running until user logon is complete.
- **Wait for <n> seconds before script timeout** - Allows you to specify the number of seconds to allow a script to continue running before the script times out. A setting of zero (0) seconds prevents the script timeout. If a timeout occurs the result is fail and settings cannot be applied.

VBScripts

Each script is run within a hosted script engine allowing greater control over the script execution whilst providing a high degree of input and output control.

- No VBS file is used.
- No separate process is spawned.

A script must be written as a function and can contain many functions, but a main start function must be specified. The start function is run by the Application Control agent and can be used to call other functions.

The AMScriptRule COM object is built into the scripting engine and provides access to the following methods:

- `strUsername = AMScriptRule.UserName`
- `strUserdomain = AMScriptRule.UserDomain`
- `strSessionid = AMScriptRule.SessionID`

- `strStationname = AMScriptRule.WinStation`



The Microsoft standard in this instance means that WinStation returns the value of the name of the Terminal Services Session, which is determined by the type of session with typical values being 'Console' or 'RDP-Tcp#34', instead of the Window Station name which is typically WinSta0.

The AMScriptRule COM object also includes the following methods:

- `strLog = AMScriptRule.Log "My Log Statement"`

Allows you to output logging strings to the agent log file for use with debugging scripted rules.

- `strEnvironmentvar = AMScriptRule.ExpandEnvironment ("%MyEnvironmentVariables%")`

Expands environment variables of the user running the script.



Using WScript. shell to expand environment variables only returns SYSTEM variables.

Windows PowerShell Scripts

If the script returns (exits) with a value of 0, the script will pass and the rules are applied. If any non-zero value is returned, the script will fail and the rules will not apply.

Each PowerShell script is executed in an instance of PowerShell.exe and as such Application Control neither enforces nor adds any specific syntax – all correctly formed PowerShell will work.



PowerShell must be installed on any endpoints that will be using the script.

Add a Scripted Rule

1. Click the **Add Rule** drop-down arrow on the Rules ribbon and select **Scripted Rule**.

A new rule is added to the All Scripted Rules work area. The **Scripted Rule** dialog displays.

2. To enter a script, do one of the following:
 - Type the script in the Current Script area.
 - Open an existing script in a script editor and copy/cut the content and paste.
3. Select **Click here to edit the script**. Click **Import** to import an existing script.

Edit a Scripted Rule

1. Use the Scripted Rule dialog to create and maintain rules based on custom VB and PowerShell Scripts that are run whenever a user logs on.

2. To open the Scripted Rule dialog for a specific rule, you can either:
 - Navigate to the scripted rule in the navigation pane and select it.
 - Select the **Rules** node in the navigation tree. In the All Rules dialog, double-click the rule that you want to edit.

The Scripted Rule dialog displays.

3. Click **Click here to edit the script.**

The Configure this Scripted Rule dialog displays.

4. In the Script tab, add or amend the script to be used when your users log on.
5. In the Options tab, select the script execution setting from the list of available options in the Define the execution settings section.
6. To specify the script time settings, select the appropriate options in the Define the script time settings section.
7. Click **OK**.

Sample scripts

Scriptable rule to determine if an AAC filter has been passed Using VBScript

The following VBscript demonstrates how to control the applications to which a user has access.

```
Function ScriptedRule()  
'Name of Filter scan expected to pass  
ExpectedFilter = "FWALL"  
'Get Server Name  
Set objNTInfo = CreateObject ("WinNTSystemInfo")  
ServerName = lcase (objNTInfo.ComputerName)  
'Set initial return value  
ScriptedRule = False  
'Create MetaFrame Session Object  
Set MFSession = Createobject ("MetaFrameCOM.MetaFrameSession")  
'Initialize the session filters for this session  
For Each x in MFSession.SmartAccessFilters  
'return true if our filter is found  
If x = ExpectedFilter Then  
ScriptedRule=True  
AMScriptRule.Log "SmartAccessFilter match found."  
End If  
Next  
End Function
```

Scriptable rule to determine if a computer is in a Computer OU Using VBScript

The following VBscript can be used to determine if a computer is in a Computer Organizational Unit:

```
Function ScriptedRule()  
ScriptedRule = vbFalse  
strCompName = AMScriptRule.StationName  
Set oRootDSE = GetObject("LDAP://RootDSE")  
strDNSDomain = oRootDSE.Get("DefaultNamingContext")  
Set oOU = GetObject("LDAP://OU=TheOUyouAreSearching,OU=Parent,OU=Parent," & strDNSDomain)  
oOU.GetInfo
```

```

For each member in oOU
    If UCase(strCompName) = UCase(member.CN) Then
        ScriptedRule = vbTrue
        Exit For
    End If
Next
End Function

```

Scriptable rule to determine if a user is a member of a certain OU Using VBScript

The following sample VBScript shows the main components of a script and demonstrates how to access information about the username of the user logging on to the system, and match with a specific domain and organizational unit:

```

Function MyScript()
'Get the username of the user logging in (also works when running as SYSTEM)
strUserName = AMScriptRule.UserName
'Get the domain of the user logging in (also works when running as SYSTEM)
strUserDomain = AMScriptRule.UserDomain
'Look up user environment variables (when running as SYSTEM, only SYSTEM variables are available)
strClientName = AMScriptRule.ExpandEnvironment ("%ClientName%")
'Log the output
AMScriptRule.Log strUserName & " logged in on " & strClientName
'Check if the user is a member of the domain
If strUserDomain = "MyDomain" Then
'If so, see if the user is in the MyOU OU
Set objOU = GetObject ("LDAP://ou=MyOU,dc=MyDomain,dc=com")
objOU.Filter = Array("user")
For Each objUser In objOU
'Check if there is a match with the user logging on
If objUser.sAMAccountName = strUserName Then
'if there is, then set the function to True
MyScript = True
End If
Next
End If
'Unless there is a username match, the function defaults to False
End Function

```

Scriptable rule to determine if a user is a member of a certain OU Using Windows PowerShell

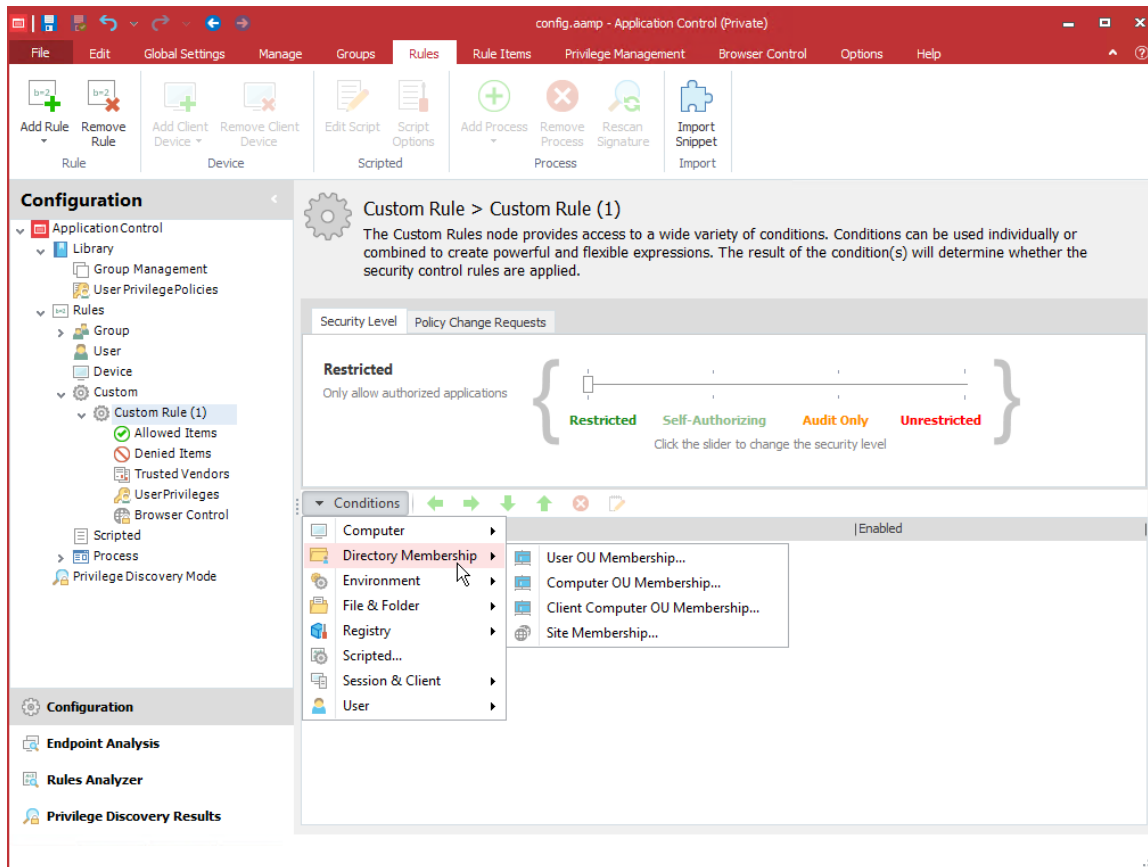
The following sample Windows PowerShell script shows the main components of a script and demonstrates how to access information about the username of the user logging on to the system, and match with a specific domain and organizational unit:

```

#Script checks if the current user is a member of the OU specified
# Return 0 if TRUE
# 1 otherwise
$loginuser = $env:username
$bindpt = [adsisearcher]"LDAP://OU=TS_Users,OU=Users,OU=MyUser,OU=MyOU,DC=MyDomain,DC=com"
$users = New-Object System.DirectoryServices.DirectorySearcher $bindpt
$users.Filter = "(&(objectClass=User)(sAMAccountName=$loginuser))"
$obj = $users.FindOne()
if($obj -eq $null)
{
#" Not a Member"
exit 1
}

```


Directory Membership Conditions



These conditions check Organizational Unit (OU) membership in Active Directory. Application Control connects to Active Directory and compares the OU specified in the condition with that of the current user or computer. If a match is made, any associated custom rules are applied. Match criteria are selected using the browse button that browses for OU containers. You must be a member of an Active Directory domain to browse for an OU container.

This condition can be used to ensure that only users in certain OUs can undertake certain actions.

Select the **Include sub-OUs in match** checkbox to search all sub-OUs of any specified OU. Without this checkbox selected, the sub-OUs are ignored and only the OU in question is included in the condition.

Condition	Description
User OU Membership	A condition based on a user's membership of a specified OU. Select whether the condition should equal or not equal the entered OU or enter a query to apply the condition to OUs.
Computer	A condition based on a computer's membership of a specified OU. Uses the same

Condition	Description
OU Membership	criteria as User OU Membership.
Client Computer OU Membership	A condition based on the membership of a specified OU for a server based or virtual client computer. Uses the same criteria as User OU Membership.
Site Membership	A condition based on the membership of a specific Active Directory Domain Site. This typically relates to an organization's departments or a geographical location which hosts networks. Environment Manager interrogates the domain to locate sites, providing them for selection from the browse button in the Match field. To browse for sites, your location must be associated with an Active Directory domain.



The OU name in the **Match** field for the **User**, **Computer** and **Client Computer OU Membership** conditions are case sensitive. OU names entered with incorrect case will not match.

Scripted Conditions

Use the Scripted condition to create, import, and export conditions using VBScript or JScript. You can use Scripted conditions to cater for scenarios that are not available as standard from the Application Control console. For example, to check if the Windows Firewall is switched on.

The scripts are held in the AAMP configuration, copied to disk at runtime, executed, and then deleted upon completion. Scripts can be imported and exported to enable reuse.



Caution: Large scripts and high numbers of scripts increase the size of an AAMP configuration, which can impact the time required to deploy configurations to endpoints and affect configuration execution time.

Because the condition scripts are run in batch mode, any prompts or message boxes are not displayed and the script times out without being executed. To ensure that a condition script runs correctly, remove or comment out any prompts or message boxes from the script.

When creating scripted conditions, make sure that there is sufficient time for the script to run and any additional conditions to be evaluated. Application Control has a timeout period of 10 seconds to evaluate any conditions for a custom rule. If all the conditions are not evaluated within 10 seconds, the custom rule is not applied. In addition, conditions are evaluated synchronously. That is, when expression evaluation is triggered, the agent waits for the script to complete before evaluating the next condition. The agent stalls the application of a custom rule until evaluation of all conditions is complete or has timed out. So even if you configure the Scripted condition to run for less than 10 seconds, if there are other conditions to evaluate and not enough time left to do it, it is still possible for evaluation to time out.

Exit Codes

All scripts for scripted conditions must specify an exit code, which when returned, is used by the Application Control agent to determine whether the script has passed or failed. For scripts without an exit code, a success (0 value) is assumed by the agent.

Each script type must use a specific exit statement:

- VBScript: **WScript.Quit** [value]
- JScript - **WScript.Quit**([value])

Replace [value] with the exit code for the script: 0 for success and 1 for failure. For example: WScript.Quit 0, WScript.Quit(0), exit (0).

Create a Scripted Condition

1. Select the node for a Custom rule.
2. In the work area for the rule, open the Conditions drop-down menu and select **Scripted**.
The Scripted Condition dialog displays.
3. Select the **Type** of scripting: **VBScript** or **JScript**.
4. In the **Run for** scroll box, set the time for which the script is allowed to run.

This is the number of seconds after which the script is terminated. The maximum value you can enter is 10 seconds. Setting the value to zero or leaving the field blank gives the script infinite time to complete. However, if the script exceeds the 10 second timeout to evaluate conditions; it times out and the custom rule is not applied.



Scripted conditions override default node and condition timeouts, but do not override the 10 second timeout to evaluate conditions. This value is hard coded

5. Click the **Options** drop-down and configure the following options as required:
 - **Evaluate Once Per Session** - Select this option to run the condition once and cache the result for the duration of the session. Otherwise, the condition is evaluated each time it is called on in a configuration.
 - **Run As System User** - Select this option to enable the script to use functionality that would not otherwise be accessible to the currently logged on user.
6. Enter the script using one of the following methods:
 - Type directly into the field
 - Drag and drop or copy and paste from another location.
 - Click the import button and select a file to open and use in the script field.
7. Click **OK** to save the script.

When triggered, the script runs to its completion and the resulting success or failure of the condition is detailed in the debug log files.

Scripts that time out are classed as failing and any child nodes and their associated actions will not run.

Export Condition Scripts

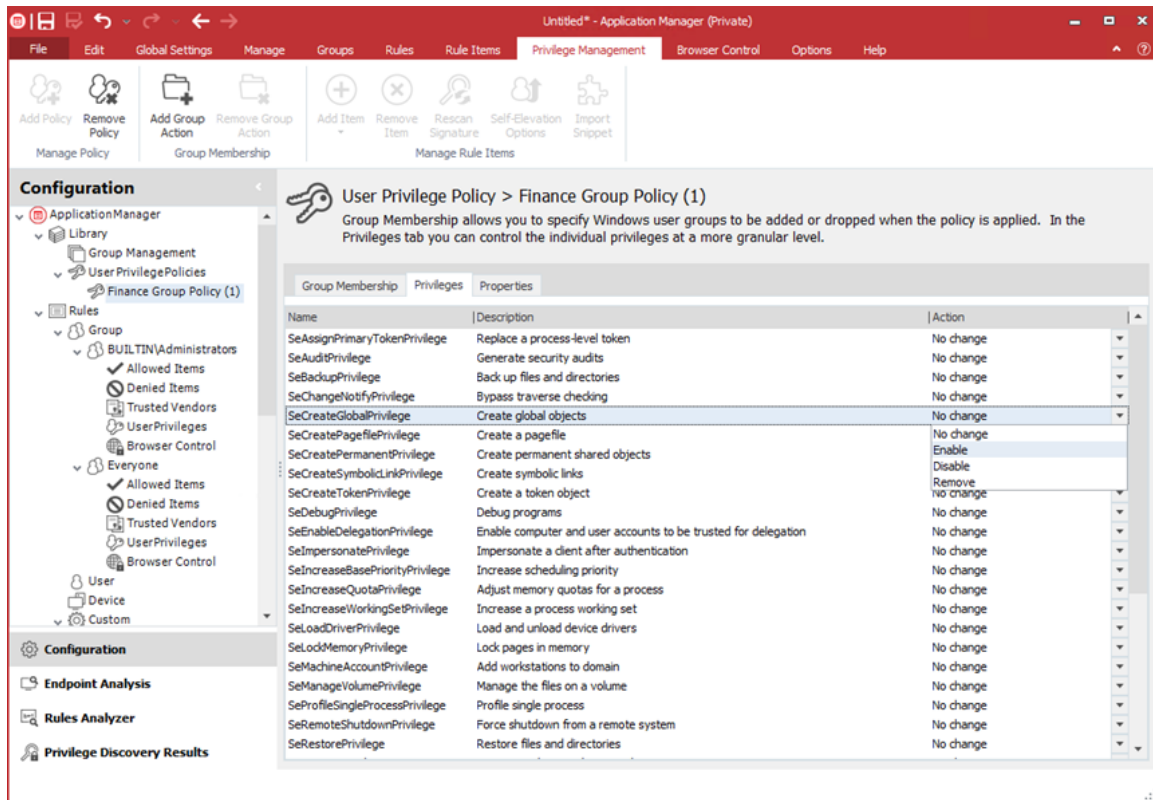
Scripts can be exported and saved from the Scripted Condition dialog and imported into other conditions and configurations.

1. Click the export button and select a location to save the file.
2. Click **Save** to complete the export.

User Privileges

A privilege is the right of a user account to perform a particular system-related operation, such as shutting down the computer or changing the system time. You can use the Privilege Management feature to assign (enable) or deny (disable) privileges.

User Privileges Policies



The Elevate policy is applied to new rule items by default. When an item is elevated the selected item will be given increased privileges and will not require an administrator to run it.

User privileges policies offer an alternative to using the default Elevate rule and can be customized to meet the needs of your organization. Policies can range from making an individual user a member of a "Power User" group to removing user membership from the Administrators group.

When a User Privileges Policy is created, you can customize your policy using the following three tabs:

- Group Membership** - Group Membership allows you to specify Windows user groups to be dropped or added when a policy is applied. You add a group action to the policy contents and then specify whether or not the selected group is to be applied to the newly created policy or whether their membership is to be dropped.

When you assign membership to a user group, you will only add the group that you have selected, any nested groups will not be included. For example, if you assign group membership to Domain Administrators this will not automatically include the Local Administrator group and they will therefore need to be added separately.

- **Privileges** - A privilege is the right of a user account to perform a particular system-related operation, such as shutting down the computer or changing the system time. You can use the User Rights Management feature to enable, disable or remove privileges:
 - **No change** - Leaves the privilege as it is with its original token.
 - **Enabled** - Sets the flag in the token to enabled.
 - **Disabled** - Sets the flag in the token to disabled.
 - **Remove** - Removes the privilege from the token. You cannot undo this option.
- **Properties** - Add a description for the policy in the Properties tab.

Create a User Privilege Management Policy

1. Select the **Library > User Privilege Policies** node.
2. Select **Add Policy** on the Privilege Management ribbon.
3. Select and right-click the new policy and select **Rename**.
4. Give the policy an intuitive name.
5. Do one or more of the following:
 - Use the Group Membership tab to specify the credentials an application can run under, for example, what group and whether to add or drop membership for the group. Adding membership allows users to run an application as if they were a member of the group.
 - Use the Privileges tab for granular control of the privileges the user will have over an application.
 - Use the Properties tab to specify the integrity level. Applications with a low integrity level cannot interoperate with applications that have a high integrity level.



User Privileges Management policies are reusable.

Add Group Membership to a Policy

Standard users typically have no administrative rights. The following process demonstrates how to create a User Privileges Policy for a Support Desk operative. User privileges management provides the ability to add membership to a selected group or to drop membership. The first step in creating the configuration is to create a User Privileges Policy and to specify the membership, in this case, to add membership.

1. In the Application Control console, select the **User Privileges Policies** node under Library.
2. In the Privilege Management ribbon, click **Add Policy**.

The new policy is added under the User Privileges Policies node in the navigation pane.

To sort policies under the User Privileges Policies node, right-click the node and select Sort Ascending or Sort Descending.

3. In the work areas, click the new policy name to make the name editable.
4. Enter a name for the policy, for example, SupportDesk.
5. In the Privilege Management ribbon, click **Add Group Action**.

The Account Selection dialog displays.

6. Enter or navigate to the SupportDesk group and click **OK**.

The group is added to the Group Membership tab in the work area for the policy.

7. In the tab, ensure that Add Membership is visible in the Action column. This is the default setting

Assign Privileges to a Policy

1. Select the **Library > User Privilege Policies** node.
2. Select **Add Policy** on the **Privilege Management** ribbon.
3. Select and right-click the new policy and select **Rename**.
4. Give the policy an intuitive name.
5. Select the **Privileges** tab for granular control of the privileges the user will have over an application.
6. Identify the privilege you want to assign.
7. Click the drop-down arrow in the **Actions** column for the privilege and select **Enable**.

Example: Create a Configuration that Allows Microsoft Silverlight to be Downloaded

Step 1 Create a Policy to Elevate to Administrator

1. Navigate to **Library > User Privilege Policies** node.
2. Select **Add Policy** ribbon button.
3. Select and right-click the new policy beneath the User Privilege Policies node and select **Rename**.
4. Enter an intuitive name for the policy, for example, *Elevate*.
5. Select **Add Group Action** ribbon button.
6. Enter the name of the administrator user group or use the Browse button to navigate to the account.
7. Ensure **Add Membership** is selected in the Action column.

Step 2 Add the Application to the User Privileges Node

1. Select **User Privileges** node for a particular group, for example, the Everyone group.
2. Select **Add Item > Application > File**.
The Add a File for User Privilege Management dialog displays.
3. Enter the name of the web installation you want to add in the File field for example *silverlight.exe* or use the Browse button to locate the file.
4. Select **Apply policy to child processes**.
5. Select **Install as trusted owner**.
6. Click **Add**.
7. Ensure the policy created in the first step procedure, *Elevate*, is selected in the User Privileges Policy column.


Step 3 Add a Signature to the Allowed Items List

1. Select the **Allowed Items** node for the same group.
2. Select **Add Item > Allowed > Signature Item**.
The Add a Signature dialog displays.
3. Navigate to the web installation and click **Open**.
4. Save the configuration.

Other configurable items also need to be considered. For example, for an ActiveX installation you need to allow the ActiveX file to run, and any executables that the control calls. You need to consider Process rules, Trusted Vendors, any Digital Certificates, Allowed Items, Elevated items, and so on.


Privileges

The following table provides the full list of privileges and describes how and when system components check for them.

Privilege	User Right	Privilege Usage
SeAssignPrimaryTokenPrivilege	Replace a process-level token	Checked for by various components, such as <i>NtSetInformationJob</i> , that set a process' token.
SeAuditPrivilege	Generate security audits	Required to generate events for the Security event log with the <i>ReportEvent</i> API.
SeBackupPrivilege	Backup files and directories	<p>Causes NTFS to grant the following access to any file or directory, regardless of the security descriptor that is present.</p> <p>READ_CONTROL</p> <p>ACCESS_SYSTEM_SECURITY</p> <p>FILE_GENERIC_READ</p> <p>FILE_TRAVERSE</p> <hr/> <p> When opening a file for the backup, the caller must specify the FILE_FLAG_BACKUP_SEMANTICS flag. Also allows corresponding access to registry keys when using.</p> <hr/>
SeChangeNotifyPrivilege	Bypass traverse checking	Used by NTFS to avoid checking permissions on intermediate directories of a multilevel directory lookup. Also used by file systems when applications register for notification of changes to the file system structure.
SeCreateGlobalPrivilege	Create global objects	Required for a process to create section and symbolic link objects in the directories of the object manager namespace that are assigned to a different session than the caller.

Privilege	User Right	Privilege Usage
SeCreatePagefilePrivilege	Create a pagefile	Checked for by <i>NtCreatePagingFile</i> , which is the function used to create a new paging file.
SeCreatePermanentPrivilege	Create permanent shared objects	Checked for by the object manager when creating a permanent object (one that does not get de-allocated when there are no more references to it).
SeCreateSymbolicLinkPrivilege	Create symbolic links	Checked for by the NTFS when creating symbolic links on the file system with the <i>CreateSymbolicLink</i> API.
SeCreateTokenPrivilege	Create a token	<i>NtCreateToken</i> , the function that creates a token object, checks for this privilege.
SeDebugPrivilege	Debug programs	If the caller has this privilege enabled, the process manager allows access to any process or thread using <i>NtOpenProcess</i> or <i>NtOpenThread</i> , regardless of the process's or thread's security descriptor (except for protected processes).
SeEnableDelegationPrivilege	Enable computer and user accounts to be trusted for delegation	Used by Active Directory services to delegate authenticated credentials.
SeImpersonatePrivilege	Impersonate a client after authentication	The process manager checks for this when a thread wants to use a token for impersonation and the token represents a different user than that of the thread's process token.
SeIncreaseBasePriorityPrivilege	Increase scheduling priority	Checked for by the process manager and is required to raise the priority of a process.

Privilege	User Right	Privilege Usage
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Enforced when changing a process's working set thresholds, a process's paged and nonpaged pool quotas, and a process's CPU rate quota.
SeIncreaseWorkingSetPrivilege	Increase a process working set	Required to call <i>SetProcessWorkingSetSize</i> to increase the minimum working set. This indirectly allows the process to lock up to the minimum working set of memory using <i>VirtualLock</i> .
SeLoadDriverPrivilege	Load and unload device drivers	Checked for by the <i>NtLoadDriver</i> and <i>NtUnloadDriver</i> driver functions.
SeLockMemoryPrivilege	Lock pages in memory	Checked for by <i>NtLockVirtualMemory</i> , the kernel implementation of <i>VirtualLock</i> .
SeMachineAccountPrivilege	Add workstations to the domain	Checked for by the Security Accounts Manager on a domain controller when creating a machine account in a domain.
SeManageVolumePrivilege	Perform volume maintenance tasks	Enforced by file system drivers during a volume open operation, which is required to perform disk checking and defragmenting activities.
SeProfileSingleProcessPrivilege	Profile single process	Checked by Superfetch and the prefetcher when requesting information for an individual process through the <i>NtQuerySystemInformation</i> API.
SeRelabelPrivilege	Modify an object label	Checked for by the SRM when raising the integrity level of an object owned by another user, or when attempting to raise the integrity level of an object higher than that of the caller's token.
SeRemoteShutdownPrivilege	Force shutdown from a remote system	Winlogon checks that remote callers of the function have this privilege.

Privilege	User Right	Privilege Usage
SeRestorePrivilege	Restore files and directories	<p>This privilege causes NTFS to grant the following access to any file or directory, regardless of the security descriptor that's present:</p> <p>WRITE_DAC</p> <p>WRITE_OWNER</p> <p>ACCESS_SYSTEM_SECURITY</p> <p>FILE_GENERIC_WRITE</p> <p>FILE_ADD_FILE</p> <p>FILE_ADD_SUBDIRECTORY</p> <p>DELETE</p> <hr/> <p> When opening a file for the backup, the caller must specify the FILE_FLAG_BACKUP_SEMANTICS flag. Also allows corresponding access to registry keys when using.</p> <hr/>
SeSecurityPrivilege	Manage auditing and security log	Required to access the SACL of a security descriptor, read and clear the security descriptor, read and clear the security event log.
SeShutdownPrivilege	Shut down the system	This privilege is checked for by <i>NtShutdownSystem</i> and <i>NtRaiseHardError</i> , which presents a system error dialog box on the interactive console.
SeSyncAgentPrivilege	Synchronize directory service data	Required to use the LDAP directory synchronization services and allows the holder to read all objects and properties in the directory, regardless of the protection on the objects and properties.

Privilege	User Right	Privilege Usage
SeSystemEnvironmentPrivilege	Modify firmware environment variables	Required by <i>NtSetSystemEnvironmentValue</i> and <i>NtQuerySystemEnvironmentValue</i> to modify and read firmware environment variables using HAL.
SeSystemProfilePrivilege	Profile system performance	Checked for by <i>NtCreateProfile</i> , the function used to perform profiling of the system. This is used by the Kernprof tool, for example.
SeSystemtimePrivilege	Change the system time	Required to change the time or date.
SeTakeOwnership	Take ownership of files and other objects	Required to take ownership of an object without being granted discretionary access.
SeTcbPrivilege	Act as part of the operating system	Checked for by the security reference monitor when the session ID is set in a token, by the Plug and Play manager for Plug and Play event creation and management, <i>BroadcastSystemMessageEx</i> when called with
SeTimeZonePrivilege	Change the time zone	Required to change the time zone.
SeTrustedCredManAccessPrivilege	Access credential manager as a trusted caller	Checked by the credential manager to verify that it should trust the caller with credential information that can be queried in plain text. Is only granted to Winlogon by default.
SeUndockPrivilege	Remove computer from a docking station	Checked for by the user-mode Plug and Play manager when either a computer undock is initiated or a device eject request is made.
SeUnsolicitedInputPrivilege	Receive unsolicited data from a terminal device	This privilege is not currently used by Windows.

User Privilege Management

Many user environments are very restrictive in order to limit user access to sensitive data and key applications. However, users often require administrative privileges to perform their role. For example, the many proprietary systems, system updates, and applications that allow the installation of drivers for devices such as printers, antivirus scans, and so on all require administrative privileges. So users typically have full administrative privileges or no administrative privileges at all.

Application Control secures and protects many corporate desktops by controlling application and network access. Application Control 8.1 and higher extends policy management capabilities by providing comprehensive user privilege management functionality. User privilege management allows you to create reusable user privilege policies that can be associated with any rules and can elevate or restrict access to files, folders, drives, signatures, Windows Store Apps, application groups, and supported Control Panel components specific to an operating system.

User privilege management enables enterprise IT departments to reduce access control privileges on a per user, group, application, or business rule basis. It ensures users have only the rights they need to fulfill their job and access the applications and controls they require, and nothing else, thus ensuring desktop stability, and improving security and productivity. The perfect balance between user productivity and security is to control user privileges, not at a session or account level, but at the level of an application or individual task.

With user privileges management, access to applications and tasks is managed dynamically by managing user privileges on demand, in response to user actions. For example, administrator privileges can be applied to a named application or Control Panel component for a particular user or user group by either elevating the privileges of a standard user to an administrator level, or dropping the privileges of an administrator to that of a standard user account.

By controlling user privileges throughout the user session, IT can provide users with the accessibility they require to perform their job, while protecting the desktop and the environment and reducing management costs.

User privileges management provides a granular approach to delegating administrative rights to users and applications by assigning rights according to merit. This level of control can be deployed to elevate or restrict privileges on a case by case basis according to the preferred approach taken in the environment.

User privileges management allows you to create a library of reusable policies that can be associated with any available Application Control rule, to assign the relevant privileges to files, folders, signatures, and application groups. User privileges policies include domain user group membership and a range of administrative privileges that you can apply to each policy.



If a new application is spawned from an existing application with administrative privileges the new application does not automatically receive the same privileges. Instead it is evaluated to determine whether or not it should receive administrative privileges.

Least Privilege

Many users run their computer with administrative privileges. Users running with these privileges can introduce viruses, malware, and spyware. This can affect an entire enterprise, causing security breaches and downtime. Access to private data can also be at risk.

User privileges management allows you to apply the principle of least privilege. This principle requires that users are provided the minimum privileges to do their job, without giving the user full administrator privileges. The experience is seamless to the user.



For the complete definition of least privilege refer to the Department of Defense Trusted Computer System Evaluation Criteria, (DOD-5200.28.STD), also known as the Orange Book. This is located at <http://csrc.nist.gov/publications/history/dod85.pdf>.

With user privileges management, any downtime, coupled with the number of calls made to IT Support due to viruses and so on, is greatly reduced because computers are made secure against the problems that occur when a user has full administrative privileges. This means IT Support can focus on more important tasks rather than spending large amounts of time troubleshooting computers to find out the problem. Licensing is also easier to control, for example, by allowing users to install only authorized applications.

Common Tasks that Require Administrative Privileges

In order to fulfil their roles, users may need to perform a number of tasks that need administrative privileges. A solution must be provided to allow these tasks to be performed; otherwise the user must fulfil their role without accomplishing these specific tasks. These tasks can include:

- Installing printers
- Installing certain hardware
- Installing particular applications
- Operating applications that require administrative privileges
- Changing system time
- Running legacy applications

User privileges management allows the user to perform these tasks by elevating a user to have specific administrative privileges.

User Privileges Management vs Run As

Many users, particularly knowledge workers, use the **Run as** command to run applications. Users can perform their daily tasks running with least privilege but can also, as required, use the **Run as** command to elevate their credentials, thus performing a task under the context of a different user. This, however, requires that a user has two accounts: one for least privileges and one for elevation.

A common problem when using **Run as** is allowing the administrative password to become known throughout an organization. For example, an administrator may communicate the administrator password to a user to enable them to use the **Run as** command to fix a problem with their computer. Unfortunately, the password commonly gets passed around, causing unforeseen security risks.

An additional problem with **Run as** is how software actually interacts with it. **Run as** executes an application or process under the context of a different user. Therefore, that application or process does not have access to the correct HKEY_CURRENT_USER hive in the registry.

This hive is where all the profile data is stored and is protected space. So the application or process running under the context of a different user cannot read or write to this source, causing some applications to not function. Running under the context of a different user can also cause problems reading and writing to a network share. This is because network shares are based on the account under the context you are running. So your local account and the **Run as** account may not have the same access to resources.

Run As and UAC

Some operating systems, such as Windows 7 and Windows 8, have features that allow a user to run applications or processes without administrative privileges. These are the **Run as** command and User Account Control (UAC). (



These features also apply to Server 2008 and 2012 versions.

Although these features do allow users to run without administrative privileges, they still require the user to have access to an administrator account to perform administrative tasks. Unfortunately, this limitation means these features are more appropriate for administrators. It enables them to log on as a standard user and use the administrator account to perform administrative tasks only.

Because the user must provide the credentials for a local administrator to use **Run as** and UAC, this creates a number of concerns. For example:

- A user with access to an administrator account must be trusted not to abuse these privileges.
- Applications running with administrative privileges are now running under the context of a different user. This can cause problems, for example, these particular applications do not have access to the actual user's profile or network shares, as stated in the User Privileges Management vs. Run As section.
- Two passwords are required. One for the standard account and one for the administrator account. The user must remember both. Security required for one account is challenging, and for two accounts more so.

Technology

In a Microsoft Windows computing environment, as part of the application launch process, when an execution request is made, the application requests a security token as part of the application launch approval process. This token details the rights and permissions given to the application and these rights can be used to interact with the operating system or other applications.

When Users privileges management is configured to manage an application, the security token that is requested is dynamically modified to have permissions elevated or restricted, thus allowing the application to be run or blocked.

1. The User Rights Management mechanism handles process startup requests as follows:
 - A User Rights Policy is defined in the configuration rule and applies to applications or components.
 - The Application list can include files, folders, signatures or application groups.
2. The Components list can include Control Panel components.
3. When a process is created by the launch of an application or other executable, the Application Control hook intercepts the process and queries the Application Control agent whether elevated or restricted rights are required to run the process.
4. The agent confirms whether the configuration assigns elevated or restricted rights and if required, the agent requests a modified user token from the Windows Local Security Authority (LSA).
5. The hook receives the modified user token from the Windows LSA granting the necessary privileges. Otherwise, the process runs with the existing user token according to the definitions of the normal user rights.

Benefits of User Privilege Management

The main benefits of User Privileges Management are:

- **Discover User Applications that Require Elevated Privileges** - Use the Privilege Discovery Mode to monitor and generate reports on applications that require administrative privileges. Use the data listed in the reports to create Application Management configurations.
- **Elevation of User Privileges for Running Applications** - Use User Privileges Management to specify the applications to be run with administrative credentials. The user does not have administrative credentials but is able to run the application.

For more information, see [Elevate User Privileges for Running Applications](#).

- **Elevation of User Privileges for Running Control Panel Applets** - Many roaming users need to do various tasks that need administrative privileges. For example, to install printers, to change network and firewall settings, change the time and date, and to add and remove programs. All of these tasks require certain components to run as administrator. Use User Privileges Management to elevate privileges for individual components so that the non-administrative standard user can make the changes to perform their role.

For more information, see [Elevate User Privileges for Running Control Panel Applets](#).

- **Reducing Privileges to Restrict Application Privileges** - By default, users have certain administration credentials, but are enforced to run specific application as a non-administrator. By running certain applications as an administrator, for example, Internet Explorer, the user is able to change many undesirable settings, install applications and potentially open up the desktop to the Internet. Use User Privileges Management to restrict an administrator level user from running, for example, Internet Explorer in a standard user mode, thus safe-guarding the desktop.

For more information, see [Reduce Privileges to Restrict Application Privileges](#).

- **Reducing Privileges to Restrict Access to System Settings** - Use User Privileges Management to give a higher level system administrator the ability to stop an administrative user from altering settings that they should not change, for example, firewalls and certain services. Use User Privileges Management to reduce administrative privileges for certain processes. Although the user has administrative privileges, the system administrator retains control of the environment.

For more information, see [Reduce User Privileges for Running Components](#).

Use Cases

User privileges management has many use cases and solves problems that many enterprises have until now been unable to address. A small number of scenarios are given below:

- Organizations that use local administrator accounts for their users may need to lock down elements of the desktop, such as the Control Panel component, Add Hardware, or Add and Remove Programs \ Programs and Features. By dynamically dropping the user account from administrator to a standard user for specific controls, the user is now prohibited from accessing the control and executing an unwanted task.
- Some applications require administrator rights because the application itself interacts with certain parts of the desktop operating system or registry. However, the organization does not wish to provide users with full administrator accounts. User privileges management can elevate the user rights for the named application to an administrator level, enabling the user to run their application while protecting the desktop.
- Automatic update elements of some applications can require administrator rights to perform the update actions and therefore not function in the context of a standard user. User privileges management can enable the named application to run under the context of an administrator account while all other applications remain in standard user context.
- Mobile users may need to manually change their IP address, configure a wireless network, or change date and time properties, all of which require administrative rights.
- User privileges management can elevate the user rights to administrator level for named tasks, enabling the user to make the changes they require.

Elevate User Privileges for Running Applications

Users often require administrative rights to perform their role. User Privilege Management allows you to elevate a user so that they have administrative rights for specified applications. To elevate user privileges, you must first create a policy and then apply this to a rule.

Step 1 - Create a User Privilege Management Policy

1. Navigate to the **Library > User Privilege Policies** node.
2. On the Privilege Management ribbon select **Add Policy**.
3. Select and right-click the new policy and select **Rename**.
4. Give the policy an intuitive name, for example, *Elevate Admin Rights*.
5. Select the new policy and in the Privilege Management ribbon click **Add Group Action**.

The Account Selection dialog displays

6. Browse to and select the group you want to add to the policy.
7. The group is listed in the Group Membership tab in the User Privilege Policy work area.

Ensure that **Add Membership** is specified in the Action column. This allows users to run an application as if they were a member of the group.

- The Group Membership tab is used to specify the credentials an application can run under.
- The Privileges tab provides granular control of the privileges the user will have over an application.
- The Properties tab is used to specify the integrity level. Applications with a low integrity level cannot interoperate with applications that have a high integrity level.

Step 2 - Apply the Policy to the Everyone Rule

1. Select **Rules > Group > Everyone > User Privileges** in the navigation pane.
2. On the Privilege ribbon select the **Add Item** drop-down arrow, highlight **Application** and then select one of **File**, **Folder**, **Signature** or **Group**.
3. Select the item you want to add.
4. Set the **User Privilege Policy** to the policy created in the *Create a User Privilege Management Policy* step above.
5. Select the **Everyone** node.
6. Move the Security Level slider to **Unrestricted** to prevent Application Control from blocking.
7. Save the configuration.



Event 9018 audits when the user privileges to an application change.

Example: Allow Users to Run Visual Studio and Debug Applications

Users often require administrative privileges to run, for example, Visual Studio, and to debug applications. Use user privilege management to elevate administrative rights for the specified applications.

To elevate user privileges, you need to first create one or more reusable policies and apply these to a rule.

Step 1 - Create a Policy to Elevate User Privileges

1. Select the **Library > User Privilege Policies** node.
2. Select **Add Policy** on the Privilege Management ribbon.
3. Select and right-click the new policy and select **Rename**.
4. Enter an intuitive name for the policy, for example, *Elevate Visual Studio*.
5. In the Privilege Management ribbon, click **Add Group Action**.
The Account Selection dialog displays.
6. Browse to and select the group you want to add to the policy.
The group is added to the Group Membership tab in the rule work area.
7. In the tab, ensure that Add Membership is specified in the Action column.
This allows users to run an application as if they were a member of the group.

Step 2 - Create a Policy to Set Privileges for Debugging

1. Select the **Library > User Privilege Policies** node.
2. Select **Add Policy** on the Privilege Management ribbon.
3. Select and right-click the new policy and select **Rename**.
4. Enter an intuitive name for the policy, for example, *Run Debug*.
5. Select the Privileges tab.
The **Privileges** work area displays.
6. Select the drop-down menu for the debugging privilege in the Action column and select **Enable**.

Step 3 - Create a Group Rule

1. Select **Rules > Group** in the navigation pane.
2. Select the **Add Rule** drop-down arrow on the Rules ribbon and select **Group Rule**.
The Add Group Rule dialog displays.

3. Enter the domain name into the Account field.
4. Click **Add**.

Step 4 - Apply the Elevate Visual Studio Policy to the Rule

1. Select the **User Privileges** node beneath the rule you have created.

The User Privilege work area displays.

2. In the Privileges Management ribbon, select **Add Item > Application > File**.
The Add a File for User Privilege Management dialog displays.
3. Browse to and select the visual studio application file.
4. Select the **Apply policy to child processes** option.
5. Click **Add**.

The file path and name of the executable file is added to the Applications tab in the work area.

6. In the tab, select the **Elevate Visual Studio** policy in the User Privileges Policy column. This is the policy created in Step 1.

Step 5 - Apply the Run Debug Policy to the Rule

1. In the User Privileges work area, select **Add Item > Application > File** from the Privileges Management ribbon.

The Add a File for User Privilege Management dialog displays.

2. Enter * in the File field. This is to allow for all debug applications.
3. Click **Add**.
4. Select the **Run Debug** policy in the User Privileges Policy column.

This is the policy created in Step 4.

Step 6 - Save

Save the Configuration.

Elevate User Privileges for Running Control Panel Components

Many roaming users need to do various tasks that need to be run as an administrator, for example:

- To install printers
- To change network and firewall settings
- To change the time and date
- To add and remove programs.

All of these tasks require components to run as administrator.

Use user privilege management to elevate privileges for individual components so that the non-administrative standard user can make the changes to perform their role.

Elevate privileges for a Component

1. Select the **User Privileges** node beneath the applicable Rules node, for example, the **Group > Everyone** node.
2. On the Privilege Management ribbon select **Add Item > Add Component**.
The Select Components dialog displays.
3. Select one or more components that you want to elevate, and click **OK**.



Use the filter at the top of the Select Components dialog to filter components by operating system.

The component is now listed on the Components tab in the policy work area.

4. Ensure the Builtin Elevate policy is selected in the User Privilege Policy column.
5. Save the configuration.

Example: Allow Users to Defragment Disks

The ability to defragment a disk requires administrative privileges and is governed by a particular component. Use privileges management to elevate user privileges for this component, thus allowing them to defragment a disk.

1. Select the applicable Rules node, for example, the **Group > Everyone** node.
2. In the Privileges Management ribbon, select **Add Item > Add Component**.

The Select Components dialog displays.

3. Select the Defragment component, and click **OK**.



Use the filter at the top of the Select Components dialog to filter components by operating system.

The component is added to the Components tab in the work area for the rule.

4. Select the drop-down arrow in the User Privileges Policy column, and select the **Builtin Elevate** policy.
5. Save the configuration.

Example: Allow Users to Perform Windows Update

The ability to update Microsoft Windows is governed by a particular component. Use privilege management to elevate privileges for this component so that the non-administrative standard user can make the changes to perform their role.

To elevate privileges for the applet:

1. Select the applicable Rules node, for example, the **Group > Everyone** node.
2. In the Privileges Management ribbon, select **Add Item > Add Component**.

The Select Components dialog displays.

3. Select the Automatic Update\Windows Update component, and click **OK**.



Use the filter at the top of the Select Components dialog to filter components by operating system.

The component is added to the Components tab in the work area for the rule.

4. Select the drop-down arrow in the User Privileges Policy column, and select the **Builtin Elevate** policy.
5. Save the configuration.

Reduce Privileges to Restrict Application Privileges

Running applications as an administrator enables a user to change many undesirable settings, install applications, and potentially open up the desktop to the Internet. Use user privilege management to restrict an administrator level user to running, for example, Internet Explorer in a standard user mode, thus safe-guarding the desktop.

To elevate user privileges, you need to first create a policy and then apply this to a rule.

Step 1 - Create a User Privilege Management Policy

1. Navigate to the **Library > User Privilege Policies** node.
2. On the Privilege Management ribbon select **Add Policy**.
3. Select and right-click the new policy and select **Rename**.
4. Give the policy an intuitive name, for example, *Reduce Admin Privileges*.
5. Select the new policy and on the Privilege Management ribbon select **Add Group Action**.
The Account Selection dialog displays

6. Browse to and select the group you want to add to the policy. These are the account credentials to run the application. Click **Add**.

The group is listed in the Group Membership tab of the policy work area.

7. Select **Drop Membership** in the Action column.
 - The Group Membership tab is used to specify the credentials an application can run under.
 - The Privileges tab provides granular control of the privileges the user will have over an application.
 - The Properties tab is used to specify the integrity level. Applications with a low integrity level cannot interoperate with applications that have a high integrity level.

Step 2 - Apply the Policy to the Everyone Rule

1. Navigate to the **Rules > Group > Everyone > User Privileges** node.
2. On the Privilege Management ribbon select the **Add Item** drop-down arrow point to **Application** and then select one of the following:
 - File
 - Folder
 - Signature
 - Group
3. Select the item you want to add.
4. Set the User Privileges Policy to the policy created in the Step 1.

5. Select the **Everyone** node.
6. Move the Security Level slider to **Restricted**.
7. Save the configuration.



Event 9018 audits when the user privileges to an application change.

Reduce User Privileges for Running Components

Use user privilege management to reduce privileges for individual components so that the non-administrative standard user cannot make certain changes.

Reduce Privileges for a Component

1. Select the **User Privileges** node beneath the applicable Rules node, for example, the **Group > Everyone** node.

2. On the Privilege Management ribbon select **Add Item > Add Component**.

The Select Components dialog displays.

3. Select one or more components that you want to reduce privileges for, and click **OK**.



Use the filter at the top of the Select Components dialog to filter components by operating system.

The selected component now displays on the Components tab in the work area.

4. Select the drop-down arrow in the User Privileges Policy column and select the **Builtin Restrict** policy.
5. Save the configuration.

Example: Restrict Users from Starting and Stopping Services

Services is a Control Panel component. Use user privilege management to reduce privileges for the Services component so that the non-administrative standard user cannot start and stop Services.

1. Select the **User Privileges** node beneath the applicable Rules node, for example, the **Group > Everyone** node.

2. On the Privilege Management ribbon select the **Add Item > Add Component**.

The Select Components dialog displays.

3. Select the **Services** component, and click **OK**.



Use the filter at the top of the Select Components dialog to filter components by operating system.

The selected component is now displayed in the Components tab in the work area.

4. Select the drop-down arrow in the User Privilege Policy column and select the **Builtin Restrict** policy.
5. Save the configuration.

Secure Dialogs

An administrator can use Application Control and Privilege Management to elevate a standard user to have administrative privileges. Allowing a user to have administrative privileges grants them access to all files, including important system files, and the ability to, for example, delete or rename them. These actions can compromise a system.

Application Control and Privilege Management provides a Secure Common Dialogs feature prohibiting users from manipulating files. The dialog boxes still open and provide access to files but the files cannot be deleted or renamed.

Application Control does not restrict access to areas that a user ordinarily has access to.

Elevate to Administrator and Secure Common Dialogs

Add a File for User Privilege Management

Properties Metadata

File

C:\Program Files\Windows NT\Accessories\wordpad.exe

☒ Substitute environment variables where possible

☐ Use regular expression

☐ Make file an Allowed Item

☐ Allow file to run even if it is not owned by a trusted owner

Arguments

Optionally specify command line arguments

☐ Use regular expression

Policy

Builtin Elevate

☐ Apply to child processes

☒ Apply to common dialogs

☐ Install as trusted owner

Description

Enter notes for your future reference

Add Cancel

Scenario

- You are an IT Administrator
- You are creating a new User Privilege policy

Process

1. Navigate to the **Library > User Privileges Policies** node and select **Add Policy** on the **Privilege Management** ribbon.
A new policy is created.
2. Right-click the new policy and select **Rename**.
3. Enter an intuitive name for the policy, for example, *Elevate to Administrator*.
4. Select the new policy and select **Add Group Action** on the Privilege Management ribbon.
The Account Selection dialog displays.

5. Type the administrator account into the Account field or use the **Browse** button to search for an account. Click **Add**.
6. Ensure **Add Membership** is selected in the Action column. This is the default setting. The Add Membership option allows users to run an application as if they were part of the specified group. The Drop Membership option does not allow the users to run an application.
7. Select the **User Privileges** node for the applicable group, for example, the Everyone group.
8. On the Privilege Management ribbon, select **Add Item > Application > File**.
9. The **Add a File for User Privilege Management** dialog displays.
10. Enter the name of the application that you want to secure common dialogs for or click the **Browse** button and browse to the application.
11. Ensure that the **Apply to common dialogs** option is selected. This is selected by default.
12. Click **Add**.
13. Ensure the policy created in steps 1 to 6 is selected in the User Privilege Policy column.
14. Save the configuration.

System Controls

Use System Controls to control the removal or modification of applications and processes, the management of specific services and the clearing of named event logs. Controls can be applied to elevate or restrict access to the specified item.

System Controls are available on User Privilege node for each rule group.

Add System Controls

1. Select the User Privileges node for the required rule group.
2. Select the **System Controls** tab
3. Click **Add Item** and select the required system control:
 - [Uninstall](#)
 - [Service](#)
 - [Event Log](#)
 - [Process Termination](#)



For Process Rules, only Process Termination items can be added.

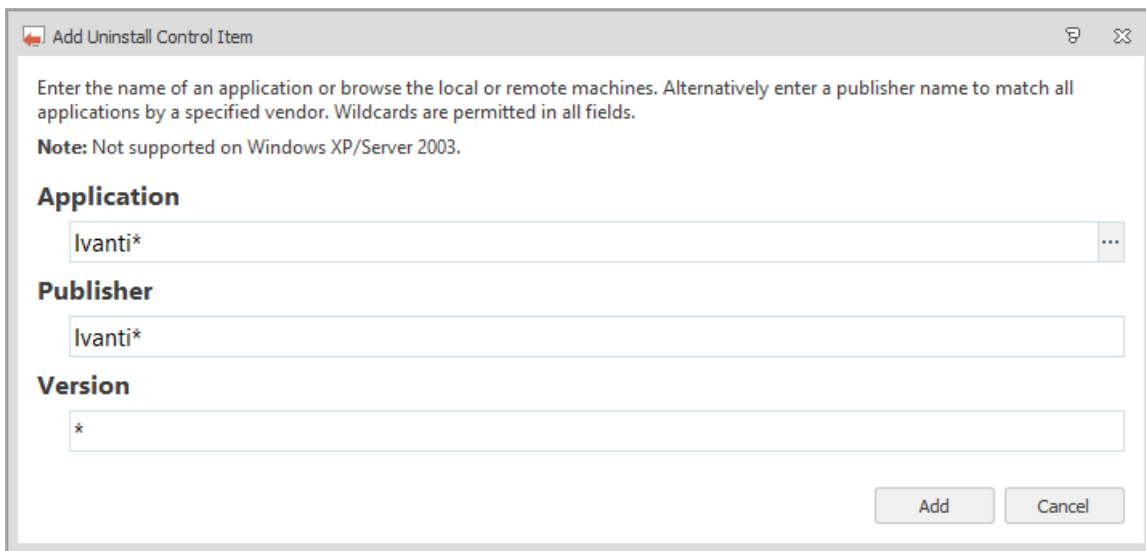
4. Complete the required details and click **OK**.
5. From the policy column, select the required option:
 - **Builtin Restrict** - Always restrict access to the named item.
 - **Builtin Elevate** - Always allow access to the named item.

6. Configure further items by clicking **Add Item** in the Privilege Management ribbon and selecting the required option.
7. Click the **Click here to set the messages displayed when a user is restricted from accessing system controls** link to configure the messages that display when a user attempts to uninstall a restricted program or clear a restricted event log.

For further information, see [Message Settings](#).

Click the **Add Ivanti Components and Dependencies** button to automatically pre-populate the Uninstall, Service, and Event Log control items configurations with an **AppSense***. Other required dependencies are also added.

Uninstall Control Items



i Application Access Control must be enabled before configuring an Uninstall Control Item. For further information, see [Policy Settings](#).

Use this option to allow or restrict installed applications from being uninstalled when the rule conditions have been matched. Uninstall Control Items are configured by defining which applications are controlled. Further validation can be applied to target a named publisher and specific application versions. To allow or restrict all applications from a publisher, enter a * in the Application field combined with the publisher name.

For an example of using the Uninstall Control Item, see [Elevating a Group to Allow Microsoft Silverlight to be Uninstalled](#).

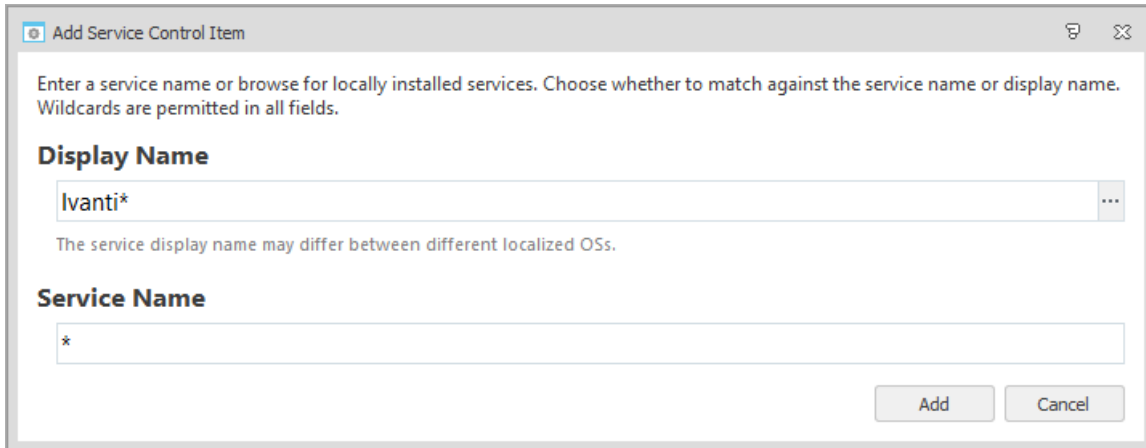
i Uninstall Control Items are not supported on Windows XP or Windows Server 2003.

AppSense Uninstall Control Items

The AppSense company name has changed to Ivanti and because of this, Application Control configurations that include AppSense* uninstall items will not control Ivanti products that previously used the AppSense name.

When a configuration that contains AppSense* uninstall control items is opened in a 10.1 FR1 (or later) console, a dialog displays. The dialog offers an option to automatically add Ivanti* to all rules that contain AppSense*. Uninstall controls function by matching the name and publisher that appear in Programs and Features. Agreeing to the automatic update ensures that, when the configuration is deployed to a rebranded Application Control 10.1 FR1 (or later) agent, AppSense products that have been rebranded to Ivanti remain subject to the same uninstall controls.

Service Control Items



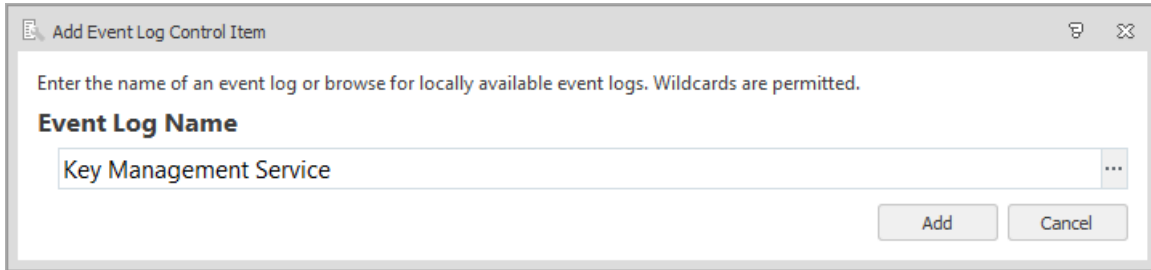
Use this option to select which services can be modified, stopped, started and restarted when the Rule conditions have been matched.

i The Agent Service is the only service that cannot be restarted once stopped.

Service Control Items are configured by specifying the service name or the name by which the service is known. The service display name may differ between different localized Operating Systems.

For an example of using the Service Control Items, see [Preventing the Windows Firewall Service from Being Stopped](#).

Event Log Control Items

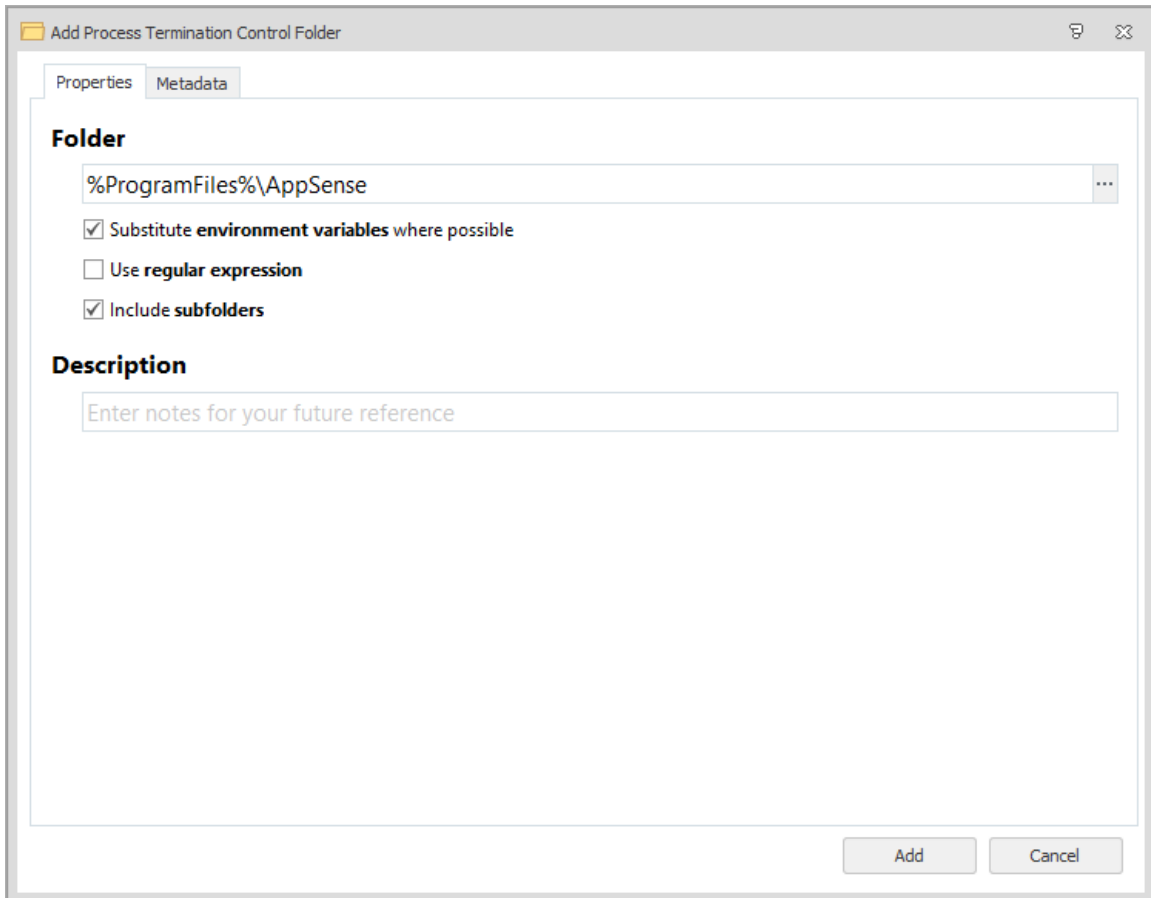


The dialog box is titled "Add Event Log Control Item". It contains a text input field labeled "Event Log Name" with the text "Key Management Service" entered. Below the input field are two buttons: "Add" and "Cancel".

Use this option to select which event logs can or cannot be cleared when the Rule conditions have been matched. Event log control items are configured by selecting the name of the log or logs to be controlled.

For an example of using the Event Log Control Items, see [Preventing the System Log from being cleared](#).

Process Termination Control Item



The dialog box is titled "Add Process Termination Control Folder". It has two tabs: "Properties" and "Metadata". The "Properties" tab is active. It contains a text input field labeled "Folder" with the text "%ProgramFiles%\AppSense" entered. Below the input field are three checkboxes: "Substitute environment variables where possible" (checked), "Use regular expression" (unchecked), and "Include subfolders" (checked). Below these checkboxes is a text input field labeled "Description" with the placeholder text "Enter notes for your future reference". At the bottom right are two buttons: "Add" and "Cancel".

Use this option to protect processes, such as antivirus software from termination by all users, including administrators. Users can still stop processes gracefully, for example, by clicking close in an application UI, but they cannot forcibly terminate a process, such as ending a task from the Details tab in Task Manager. An individual file can be specified or all processes in a particular folder can be targeted.

Optionally, add Metadata to include additional criteria for matching files and folders.

For further information, see [Metadata](#).

Elevate a Group to allow Microsoft Silverlight to be uninstalled

Scenario

You are an IT Administrator

You are creating an Application Control configuration

You have created a Corporate\ITSupport-Level 1 group

You want to elevate the Corporate\ITSupport-Level 1 group to allow them to uninstall Microsoft Silverlight

Process

1. In the Corporate\ITSupport-Level1 group rule, select the **User Privileges** node.
2. Select the **System Controls** tab.
3. In the work area, right-click and select **Uninstall Control Item**.

The Add Uninstall Control Item dialog displays.

4. Use the ellipsis in the Application field to navigate to the Browse Installed Applications dialog and select **Microsoft Silverlight**.



Alternatively, enter the name of the application in the field provided. Wildcards can be used if required. For example, to specify Microsoft Silverlight, you might input ***silverlight**.

For further information on wildcards, see [Wildcards and Regular Expressions](#).

If the application to uninstall is located on another endpoint, click the **Connect** button and enter the endpoint name and your credentials. Select the required application from the list. To select more than one application, hold down the **Ctrl** button on your keyboard and select the required applications.

If you are connected remotely to another endpoint, click the **My Computer** button to view the list of installed applications on your local machine.

5. Click **Add**.

The Application, Publisher and Version details are automatically populated in the Add Uninstall Control dialog.



To apply the control item to all versions of Microsoft Silverlight, replace the version number with a *.

6. Click **Add**.
7. Select **BuiltIn Elevate** from the drop-down list in the Policy Column.

By selecting the **BuiltIn Elevate** option, you are granting the application or component the privileges to complete a specific action that would otherwise need to be performed by an Administrator.

Alternatively, select the **BuiltIn Restrict** option to restrict the application or component from automatically completing the action.

When any user within the Corporate\ITSupport-Level1 group attempts to uninstall a version of Microsoft Silverlight, both the Publisher name and the Version number must match before the uninstallation action can be performed. If the criteria for the application does not match, users will be prevented from completing the action, unless an Active Directory Policy dictates otherwise.

Prevent the Windows Firewall Service from being stopped

Scenario

- You are an IT Administrator
- You are creating an Application Control configuration
- You want to prevent the Windows Firewall service from being stopped by everyone in the organization

Process

1. In the Everyone group, select the **User Privileges** node.
2. Select the **System Controls** tab.
3. In the work area, right-click and select **Service Control Item**.
4. The Add Service Control Item dialog displays.

5. Use the ellipsis in the Display Name field to navigate to the Browse Installed Services dialog and select **Windows Firewall**. Alternatively, enter the name of the service in the field provided. Wildcards can be used if required.

For further information on wildcards, see [Wildcards and Regular Expressions](#).

If the service is located on another endpoint, click the **Connect** button and enter the endpoint name and your credentials. Select the required service from the list of installed services. To select more than one service, hold down the Ctrl button on your keyboard and select the required services.

If you are connected remotely to another endpoint, click the **My Computer** button to view the list of services on your local machine.

6. Click **Add**.

The Display Name and Service Name details are automatically populated in the Add Service Control Item dialog.

7. Click **Add**.
8. Select **Builtin Restrict** from the drop-down list in the Policy Column. By selecting this option, you are granting the application or component the privileges to complete a specific action that would otherwise need to be performed by an Administrator. Alternatively, select the **Builtin Restrict** option to restrict the application or component from automatically completing the action.
9. Users are prevented from stopping the Windows Firewall service.

Prevent the System Log from being cleared

Scenario

- You are an IT Administrator
- You are creating an Application Control configuration
- You want to create a Corporate\ITSupport-Level 2 group rule
- You want to prevent members of this group from clearing the System log

Process

1. Select the **Group Rules** node.
2. In the work area, right-click and select **Add Group Rule**.
3. In the Add Group Rule dialog, enter **Corporate\ITSupport-Level2** and click **Add**.
The ITSupport-Level2 group is created.
4. Select the **User Privileges** node.
5. Select the **System Controls** tab.

6. Right-click in the work area and select **Event Log Control Item**.

The Add Event Log Control Item dialog displays.

7. Use the ellipsis to navigate to the Browse Installed Event Logs dialog.
8. Select **System** from the event log list and click **Add**. To select more than one log, hold down the **Ctrl** button on your keyboard and select the required logs.
9. On the Add Event Log Control Item dialog, click **Add**.

The System event log is added as a controlled item.

10. Select **BuiltIn Restrict** from the drop-down list in the Policy Column. This option lowers the privileges of the users within the group. By restricting users in this way, you are preventing them from performing administrative actions such as clearing event logs. Alternatively, select the **BuiltIn Elevate** option to grant access to clear the event logs.

Members of the Corporate\ITSupport-Level2 group are prevented from clearing the System event log.

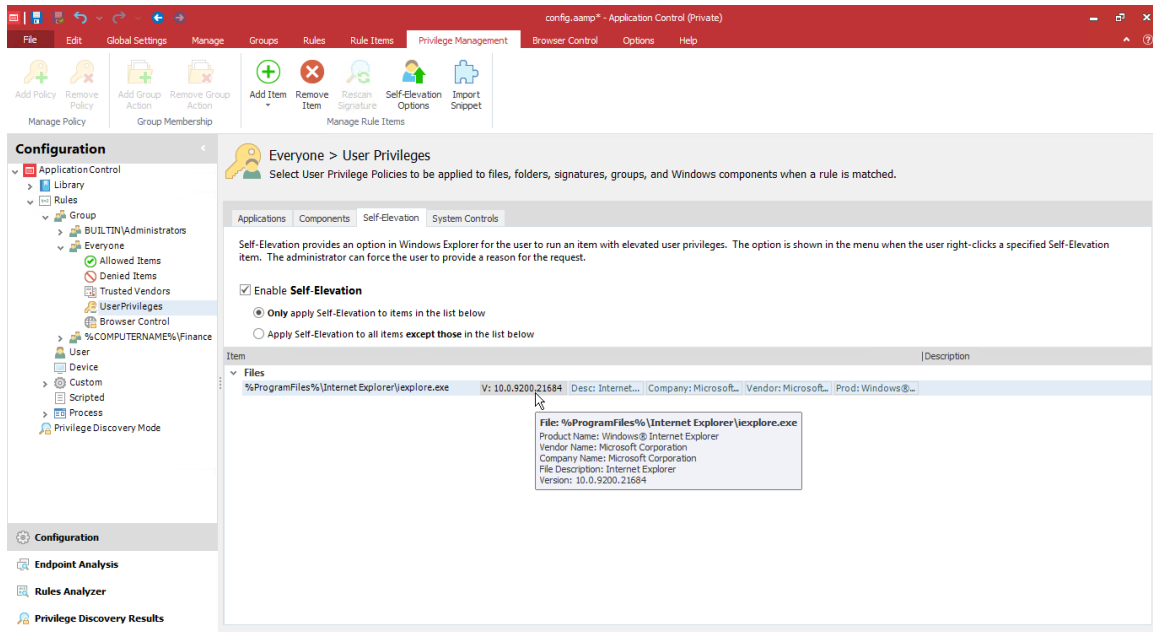
Self-Elevation

Self-Elevation can be applied to signatures, files, and folders that usually require administrative privileges to run and function. Self-Elevation provides an option from the Windows Explorer shortcut menu to run an item with elevated rights. When a user attempts to elevate a specified item, a prompt displays to request that the user enters a reason for the elevation before it is applied.

Self-Elevation is audited so you can monitor the types of applications that users typically want to self-elevate. You can add these items to the appropriate User Privileges node in a configuration so users can access them without request.

In environments where User Access Control (UAC) is disabled, you can enable the self-elevation of Windows Explorer file and folder properties using the custom setting, `SelfElevatePropertiesEnabled`. In this case, you can customize the Windows Explorer shortcut menu option text using the custom setting, `SelfElevatePropertiesMenuText`.

Configure Self-Elevation



1. Select the **User Privileges** node for the applicable group, for example, the *Everyone* group.
2. Select the **Self-Elevation** tab.
3. Select **Enable Self-Elevation** and apply the required setting:
 - **Only apply Self-Elevation to items in the list below**
 - **Apply Self-Elevation to all items except those in the list below**
4. In the Manage ribbon, select **Add Item** > **Self-Elevation** and enter or choose a file, folder, signature, or group.



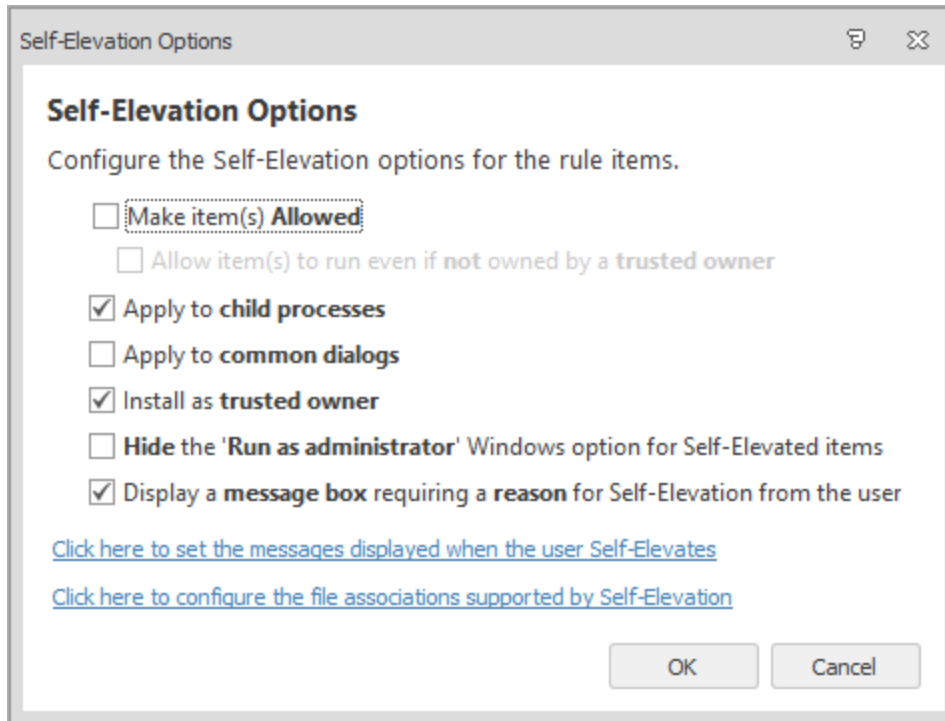
Any file type can be self-elevated if it is included on the [Self-Elevation File Associations](#) list.

5. To add further validation, click the **Metadata** tab and enter details about the description, vendor, and version number of the file and product.

Leave the fields blank if you do not want to restrict which files can be self-elevated. Where metadata has been applied, items must match that metadata to be self-elevated.

6. Save the configuration.

Self-Elevation Options



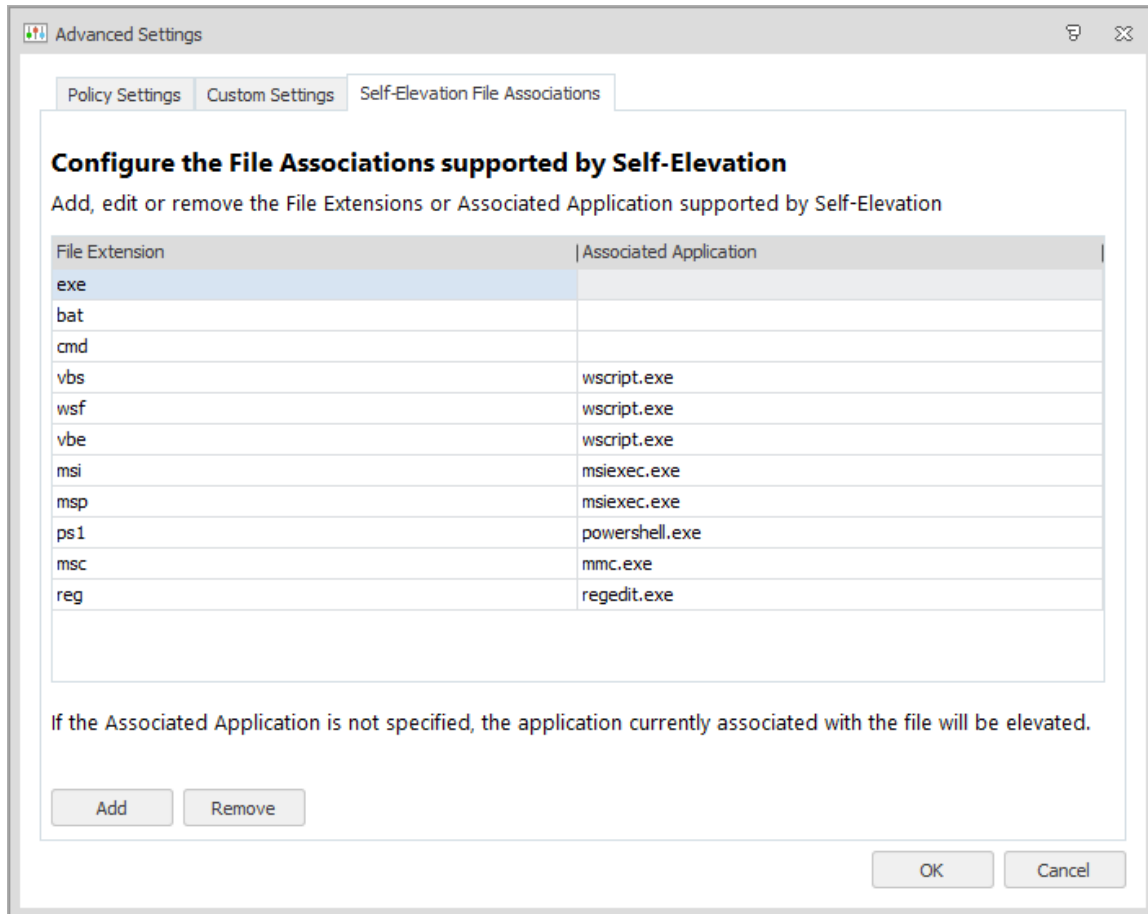
Configure the Self-Elevation options for the rule items by specifying how an application runs once it has been elevated. You can also define how the elevation is to affect any child processes or common dialogs.

In the Privilege Management ribbon, select **Self-Elevation Options** and configure the required settings:

- **Make item(s) Allowed** - Make the rule items allowed and overwrite any associated allowed items.
- **Allow items to run even if it is not owned by a trusted owner** - This option is available when Make item(s) Allowed is selected. When selected, all the rule items listed are executed regardless of the owner.
- **Apply to child processes** - By default, the Self-Elevation Policy applied to rule items is not inherited by child processes. Select this option to apply the policy to the direct children of the parent process.
- **Apply to common dialogs** - Elevate access to the Open File and Save File Windows menu options when a file or folder has been elevated. By default, any common dialogs are not elevated.
- **Install as trusted owner** - Make the local administrator the owner of all files created by the defined application. This option is not applied to regular applications, only installer packages.
- **Hide the Run as Administrator Window options for Self-Elevated items** - Hide the Run as Administrator option from the Windows shortcut menu.

- **Display a message box requiring a reason for Self-Elevation from the user** - Prompt users to provide a reason when they self-elevate. Set the content and dimensions of the message in the [Self-Elevation Message Settings](#).

Self-Elevation File Associations



Configure a list of file types and associated applications that users can open with elevated or administrative privileges. When a user right-clicks a file, Application Control performs the following checks to determine whether the user can elevate the application associated with the file:

- Is the file type on the file associations list?
 - **No** - the file cannot be self-elevated.
 - **Yes** - check the associated application.
- Is there an associated application?
 - **No** - the file is self-elevated using the application currently associated with the file.
 - **Yes** - the file can be self-elevated only if opened with the application specified in the file associations list.

If the application can be self-elevated, a corresponding option is available from the shortcut menu and the user accesses the application with elevated privileges. If a user changes a default program to one that differs to the associated application set in the configuration, the self-elevation option is no longer available from the shortcut menu.

Update Files Associations

1. In the Manage ribbon, click **Advanced Settings** and select the **Self-Elevations File Associations** tab.
2. Update the list of extensions and associated applications using the **Add** and **Remove** buttons. Any file extension can be added.

The following extensions are included by default:

File Extension	Associated Application
EXE	
BAT	
CMD	
VBS	wscript.exe
WSF	wscript.exe
VBE	wscript.exe
MSI	msiexec.exe
MSP	msiexec.exe
PS1	powershell.exe
MSC	mmc.exe
REG	regedit.exe

Self-Elevation Message Settings

The screenshot shows the 'Message Settings' dialog box with the 'Self-Elevation' tab selected. The dialog is titled 'Define the messages displayed to the user'. It contains two sections: 'Define the Self-Authorization Message users will see' and 'Define the Response Message users will see'. Each section has a 'Caption' field, a 'Message body' text area, and 'Width' and 'Height' input fields. The 'Self-Authorization' section has a caption 'Application Control - Security Warning' and a message body '%ExecutableName% cannot run without your authorization. This action may be logged.' The 'Response' section has a caption 'Application Control - Authorized File' and a message body '%ExecutableName% is now authorized. Applications using this file may need to be restarted.' Both sections have a link 'Click here to see how this message will appear to users'. At the bottom right are 'OK' and 'Cancel' buttons.

Message Settings

Define the messages displayed to the user

Message Style | Access Denied | Application Limits Exceeded | Time Limits

Self-Authorization | Network Connections | **Self-Elevation** | System Controls

Define the Self-Authorization Message users will see

Caption Application Control - Security Warning

Message body %ExecutableName% cannot run without your authorization. This action may be logged.

Width 0 Height 0

[Click here to see how this message will appear to users](#)

Define the Response Message users will see

Caption Application Control - Authorized File

Message body %ExecutableName% is now authorized. Applications using this file may need to be restarted.

Width 0 Height 0

[Click here to see how this message will appear to users](#)

OK Cancel

Configure the content and dimensions of the message that displays when a user requests self-elevation.

The messages are displayed if the *Display a message box requiring a reason for Self-Elevation from the user* option is selected in the [Self-Elevation options](#).

1. In the Global Settings ribbon, select **Message Settings**.
2. Select the **Self-Elevation** tab.
3. In the Name field, enter the text to display for the self-elevation shortcut menu option.

The menu option is displayed when a user right-clicks a file with an extension on the [Self-Elevation file associations](#) list.

4. Configure the caption, content, and dimensions for the message that displays when a user requests self-elevation.
5. Click **OK**.

Application Network Access Control

Application Network Access Control (ANAC) provides the ability to control outbound network connections by IP Address, Host name, URL, UNC, or Port, based on the outcome of the rules processing. For example, access based on location of requestor - connecting through VPN or directly to network.

Application Network Access Control is designed to control access within a company network infrastructure. This control is achieved by intercepting application requests made through the WINSOCK layer. For example,

Network Connection Items can be created individually or as part of a Group. Groups and Items can be applied to any rule in Allowed Items to allow access or in Denied Items to deny access. Application Control intercepts and blocks network access if requests are made to deny network resources. The execution of applications is not controlled.

Access is allowed to all network resources until actively denied.

Network Connection Items

Network Connection Items can be created for any network resource and can be added to a configuration in the following ways:

- **Directly to a Rule** - Adding single Network Connection Items to Allowed and Denied Item lists are advantageous when a more granular level of control is required, or when only a few items are required. However, using this method could prove time consuming.
- **Assign to Group** - Duplicate Network Connection Items are not allowed in the same Group.
- **Copy and Paste** - Network Connection Items can be cut, copied, or dragged and dropped between rules. There are no default Network Connection Items in a configuration. The full path of the Network Connection Item cannot exceed 400 characters.

Add a Network Connection

Connection Type

Select one of the following types:

- **IP Address** - Select to control access to a specific IP Address.
- **Network Share** - Select to control access to UNC paths. The prefix \\ is added to the Host field.
- **Host Name** - Select to control access to a specific Host Name.

Connection Options

The combined number of characters for all three fields, Host, Port and Path must not exceed 400.

Host

The IP Address or Host Name for the network connection. This depends on the type of connection selected. The ? and * wildcards can be used. Additionally, ranges can be used for IP Addresses, which are indicated by use of a hyphen (-).

An IP Address must be in IP4 octal format. For example, n.n.n.n

If Network Share is selected as the connection type, the \\ prefix is required.

The full path for the target resource can be entered in Host.

Example:

Enter http://server1.company.local:80/resource1/ in the Host field.

Move focus away from Host and the path is automatically split into the separate connection options:

- http:// is removed from the Host field and server1.company.local remains.
- : is removed and 80 is moved to Port.
- /resource1/ is moved to Path.

This allows a full path to be copied and pasted with ease.

Port

The port number of the network connection. This can be used in combination with IP Address or Host Name to control access to a specific port. Ranges and comma separated values are allowed as a part of the port number.

Click **Common Ports** to display a list of commonly used ports. Select as many ports as required.

Path

The path of the network connection. The ? and * wildcards can be used. To use



The Path is only relevant for controlling HTTP and

- Text contains wildcard characters - Select to use the characters ? and * as wildcards in the Path. If not selected, ? and * are treated as URL delimiters.
- Use Regular Expressions - Select this option to use regular expressions for the selected path.
- Include subdirectories - Select to include subdirectories in the rules processing.
- Only applicable if the connection type Network Share is selected.

Description

Enter a meaningful description to describe the network connection.

Add a Network Item Directly to a Rule

Network Items can be added to any Allowed Items or Denied Items node. For example, A Network Connection Item is set up for an IP Address. The Network Connection Item is assigned to Denied Items, in a Group Rule. The group members of that rule, will not have access to any network resources with that IP Address.

1. Navigate to the required node, for example, Denied Items or Allowed for a specific user group.
2. From the Rule Items ribbon, select **Add Item > Denied (or Allowed) > Network Connection Item**.

The Add a Network Connection dialog displays.

3. Fill in the details of the connection type.
4. Click **Add**.

Edit a Network Connection Directly in a Rule

1. Navigate to the Rule node in the navigation tree where the Network Connection Item to be amended is located.
2. The relevant work area displays.
3. Click on the Network Connection Item to be amended, listed under Network Connections.
4. Select **Edit Network Connection** on the Rule Items ribbon.
5. The Edit a **Network Connection** dialog displays.
6. Make the required amendments.
7. Click **OK** to save the changes and close the dialog.

Assign a Network Connection Item to a Group

1. Navigate to the **Group Management** node.
2. Select the group, to which to add the Network Connection Item, in the navigation tree.
3. Right-click within the work area and select **Add > Network Connection**.

The Add a Network Connection dialog displays.

4. Specify the Network Connection details and click **Add**.

Edit a Network Connection Item in a Group

1. Navigate to relevant Group in the navigation tree.
The Group Management work area displays.
2. Select the Network Connection Item to be amended, listed under Network Connections.
3. Select **Edit Item** on the Groups ribbon.
The Edit a Network Connection dialog displays.
4. Make the required amendments.
5. Click **OK** to save the changes and close the dialog.

Application Network Access Control and Reverse DNS Lookup

The Application Network Access Control feature can use reverse DNS lookups when evaluating Network Connection rules. The feature is turned off by default, as the time it takes to retrieve this information from DNS servers, may degrade the performance of network applications.

Enabling this feature ensures the network rules are more effective, in situations when users or applications make requests for network resources, using IP addresses when the configuration is based upon host names.

The reverse DNS lookups can be enabled by configuring a set of engineering keys.

This feature requires an administrator to enable and configure Reverse DNS Zones on the DNS servers.

For further information, refer to the *Application Control Engineering Settings Guide*.

Configure Reverse DNS Lookup Entries

If using the engineering keys to configure reverse DNS lookup entries only add IP Addresses that are within the company network infrastructure to the relevant engineering key.

Endpoint Configuration Merging

Endpoint Configuration Merging uses the Application Control Agent to combine multiple AAMP configuration files, saved on one endpoint, into a single configuration. Attributes such as group, user, custom and other rule types along with application and policy libraries from each configuration are added to the merged configuration.

The merge is done by adding the individual configurations to a directory on the endpoint and specifying, in a manifest file, the configurations which are to be merged. The Agent monitors the merge directory and automatically merges configurations when a manifest file is added to the directory.

Endpoint Configuration Merging allows different areas of a business to work independently on a particular area of a configuration, which can then be merged to create a single configuration.



System Center Configuration Manager Integration is not supported in Endpoint Configuration Merging.

Merge Components

Base Components

Every merge must have a base configuration - this is the first configuration in the merge onto which the other configurations are added. A merged configuration takes the global attributes such as Message Settings, Auditing, and any Default settings, from the base configuration.

It is therefore essential that the settings that are not merged are defined in the base.

By default, the base configuration is set as the AAMP file that is created when a live configuration is saved on an endpoint:

%PROGRAMDATA%\AppSense\Application Manager\Configuration\configuration.aamp

However, any configuration in the merge can be set as the base configuration.



Only AAMP files which are at the latest version can be included in a merge.

Component Configurations

A merged configuration is made up of a base configuration and one or more component configurations. Component configurations are AAMP files that are added to the base configuration during a merge. To be part of a merge, component configurations must be stored in the MergeConfigs directory.

MergeConfigs Directory

This directory is where component configurations for merging are stored and where a merge is triggered when a valid manifest is detected.

When you start the AMAgentService on an endpoint, the MergeConfigs directory is created:

%PROGRAMDATA%\AppSense\Application Manager\MergeConfigs



This directory is secured so only administrators can write to it. This ensures that end users cannot affect the merge configurations

ManifestGen Tool

Application Control includes a command line tool to assist when merging configurations and snippets. The ManifestGen tool creates the XML manifest file used to define and trigger a configuration merge. The XML file contains details of the AAMP files to be merged and can dictate whether the system configuration.aamp or a component configuration is used as the base in the merge.

To make using the tool easier, add its location to **Advanced System Properties > Environment Variables > Path**:

%PROGRAMFILES%\AppSense\Application Manager\Console\ManifestGen.exe

Manifests

The manifest is an XML file that includes details of the configurations to be merged and dictates whether or not the base configuration is included. The manifest initiates the merge when detected in the MergeConfigs directory - if detected by the agent, the merge begins.

Manifests are created using the ManifestGen command line tool.

The table below shows the attributes and tags which make up a manifest XML file:

Attribute/Tag	Description
MergeManifest	The root node of the configuration.
MergeFiles	The container tag for the list of AAMP files that are to be included in the merge.
FileEntry Name	Identifies a configuration to be included in the merge. The file must be present in the MergeConfigs directory to be included in a merge
UseSystemBase	UseSystemBase is set to "true" by default if not present in the manifest. It can be set to "true" or "false", and instructs to either include or exclude the default Configuration.aamp in the merge. This is the live Configuration.aamp

Attribute/Tag	Description
	<p>file found in %ProgramData%\AppSense\Application Manager\Configuration\configuration.aamp.</p> <p>If set to "true", the base configuration must already be present on endpoints when the manifest is deployed, otherwise the merge will fail.</p> <p>If set to "false" the first configuration in the MergeFiles list is used as the base configuration unless otherwise defined by the BaseConfig attribute.</p>
WaitForConfigs	<p>Determines the behavior when a manifest .xml is detected in the MergeConfigs directory and not all named configurations are present. Can be set to:</p> <p>True - This is the default setting if it is not already present in the manifest. The merge will wait indefinitely until all configurations referenced in the manifest are present and then complete the merge. This is ideal if you are deploying the manifest with the configurations and you cannot be sure in what order they will be added to the directory. For example, when you are using an MSI.</p> <p>False - The merge will fail if a manifest is detected in the MergeConfigs directory which references a configuration which is not present.</p> <p>If you are using an installer, such as an MSI, to push out configurations and a manifest to endpoints, it is recommended that you set this to "true" as you cannot guarantee in what order the configurations and manifest will be added.</p> <p>This does not apply if using the SystemBase Configuration.aamp file. If the manifest merge is triggered and the Configuration.aamp is not present, the merge will fail - it will not wait for the base</p>
Checksum (Optional)	<p>An MD5 checksum unique to an AAMP file. If included in the manifest, the AAMP file in the MergeConfigs folder must have the same checksum to be included in the merge.</p> <p>Base configurations are not referenced by a checksum.</p>

Create a Manifest

1. Save the configurations you want to be merged in the MergeConfigs directory:
%PROGRAMDATA%\AppSense\Application Manager\MergeConfigs
2. Open the Command Line Interface.
3. Enter **cd %programdata%\AppSense\Application Manager** to change the directory.

4. Enter **manifestgen "C:\ProgramData\AppSense\Application Manager\MergeConfigs*.aamp"**.

If you run manifestgen in the MergeConfigs folder, the agent picks up the manifest as soon as it is created and immediately start the merge.

If successful, a merge_manifest.xml file is created in:

%PROGRAMDATA%\AppSense\Application Manager

The manifest can now be used to trigger the merge and create a configuration.

If a merge_manifest.xml already exists in the output directory, the tool fails and a new manifest is not created - the current one is not overwritten.

Additional Commands

Suffix	Description and Usage
-o	<p>Output file - Specify where the merged configuration is generated, for example,</p> <pre>manifestGen "C:\ProgramData\AppSense\Application Manager\MergeConfigs*.aamp" -o C:\Configs</pre> <p>creates a manifest that will create a merged configuration in the Configs folder on the C drive.</p>
-b	<p>Base configuration - Identify the base configuration and exclude the system base configuration. For example,</p> <pre>manifestGen "C:\ProgramData\AppSense\Application Manager\MergeConfigs*.aamp" -b Config1.aamp</pre> <p>creates a manifest that will create a merged configuration with Config1.aamp set as the base configuration.</p>
-nc	<p>No checksum entries - By default, each configuration listed in the manifest has an MD5 checksum which allows unique identification of a configuration. If the checksum in the manifest does not match that of the configuration the merge will fail. Using the -nc suffix with the ManifestGen tool will not list checksums in the manifest and means that merges will succeed if the configuration file names are correct, regardless of the checksum value. For example:</p> <pre>manifestgen "C:\ProgramData\AppSense\Application Manager\MergeConfigs*.aamp" -nc</pre>
-nw	<p>The default behavior when a manifest is added to the MergeConfigs directory is to wait indefinitely until all configurations in the manifest are present and then perform the merge. However, the merge does not wait when a baseconfig and layer is missing and will fail immediately.</p>

Suffix	Description and Usage
	<p>Using the -nw suffix, a merge will fail if the configurations listed are not present when the manifest is added to the MergeConfigs directory. For example:</p> <pre>manifestgen "C:\ProgramData\AppSense\Application Manager\MergeConfigs*.aamp" -nw</pre> <p>If the manifest lists five configurations and only four are present when the manifest is added to the MergeConfigs directory, the merge will fail.</p> <p>If you are using an installer, such as an MSI, to push out configurations and manifests to endpoints, it is recommended that you do not use this suffix as you cannot guarantee in what order the configurations and manifests will be added.</p>

Edit a Manifest

Once created, a manifest file can be edited to change the attributes such as the base configuration and the order in which the merge should take place.

Although manifests can be edited and created in a text editor, it is recommended that you use the ManifestGen tool because it ensures the merge_manifest.xml file is in the correct format. If, for example, you have an "&" in a file name, the ManifestGen tool will escape this to make sure it is a valid XML file.

For example, the command:

```
manifestgen "C:\ProgramData\AppSense\Application Manager\MergeConfigs\*.aamp" -b
mergeconfigs\config3.aamp
```

creates a manifest in which the system base configuration is not included and config3.aamp is set as the base.

```
<MergeManifest UseSystemBase="false" WaitForConfigs="true">
```

```
  <MergeFiles>
```

```
    <FileEntry Name="config3.aamp" BaseConfig="true"
      Checksum="e899e0f9a5afee4eb502072a61c2eo" />
```

```
    <FileEntry Name="config2.aamp"
      Checksum="e899e0f9a5afef5e4eb502072ac2e0" />
```

```
    <FileEntry Name="config1.aamp"
      Checksum="b50576a6d743cf361c37b64bcaca75" />
```

```
  </MergeFiles>
```

```
</MergeManifest>
```

To edit the manifest, open the manifest in a text editor, make the required changes and save the file.

In this example, UseSystemBase is set to "true" and the BaseConfig-"true" command has been removed from config3.aamp. The order of the merge has also been changed.

```
<MergeManifest UseSystemBase="true" WaitForConfigs="true">
  <MergeFiles>
    <FileEntry Name="config1.aamp"
      Checksum="b50576a6d743cf361c37b64bcaca75" />
    <FileEntry Name="config2.aamp"
      Checksum="e899e0f9a5afef5e4eb502072ac2e0" />
    <FileEntry Name="config3.aamp"
      Checksum="e899e0f9a5afee4eb502072a61c2eo" />
  </MergeFiles>
</MergeManifest>
```

When merged, the system base configuration.aamp file is included in the merge as the base configuration and the order in which the component configurations are merged onto the base is reversed.



Setting BaseConfig="true" for a configuration and UseSystemBase="true" in the same manifest will cause a conflict and the merge will fail.

Empty Manifest

Adding an empty manifest to the MergeConfigs directory automatically merges all AAMP configurations within that directory. It will merge all configurations in alphabetical order and set the base as the configuration.aamp found in:

%ProgramData%\AppSense\Application Manager\Configuration

If this AAMP is not present, the merge will fail.

To create an empty manifest, open a new file in a text editor, create a zero-byte file and save as merge_manifest.xml.

The same merge can be achieved using a manifest that, whilst not totally empty, does not include details of the AAMP files to be merged:

```
<MergeManifest UseSystemBase="true" WaitForConfigs="true"
/>
```

This provides the same results as a blank manifest but allows you to use the UseSystemBase attribute. If you set this to "false" the merge will use the configuration which is first alphabetically in the MergeConfigs directory, as the base.

Merge Configurations

Once you have created a manifest and have your configurations on your endpoints in place for merging, you can trigger the merge and create a new configuration.

A merge is triggered when a merge_manifest.xml is detected in the MergeConfigs directory which should contain all the configurations you want to merge.

If the manifest lists configurations which are not in the MergeConfigs directory, the merge will be delayed until all configurations are present.



Using the -nw tag, a manifest can be created which will fail a merge if all configurations are not present

Successful Merges

If the manifest is correct and the configurations listed are present in the MergeConfigs directory, the merged_configuration.aamp is created and used as the live configuration on endpoints.

In addition to the new configuration (merged_configuration.aamp) a copy of the successful manifest (last_merge_manifest.xml) is added to the folder to provide a record of the merge and a backup of the manifest. The original merge_manifest.xml file is deleted when the merge is complete.

The configuration.aamp file is not altered during a merge and is no longer used by the agent unless updated or the Merged_Configuration.aamp is not present.

Unsuccessful Merges

If an error occurs during the merge, it will fail and a new configuration file will not be created. Merges can fail if:

- The checksums specified in the manifest do not match those of the actual configurations and WaitForConfigs is set to "false".
- The manifest includes the -nw command and one or more configurations listed in the manifest are not present in the MergeConfigs directory when it is added.
- Friendly names are the same in two of the configurations being merged.
- UseSystemBase is set to "true" and a base Configuration.aamp is not present when the merge is triggered.
- A manifest is invalid.
- One or more configurations are corrupt.

Following an unsuccessful merge, the merge_manifest.xml file is deleted and a copy of the unsuccessful manifest (failed_merge_manifest.xml) is added to the directory.



Further details about merging errors can be found by using Windows Event Viewer (select **Windows Log > Applications**). The event will show the reason for the failure and which configuration is causing the merge to fail.

Merge Behaviors

The table below lists the configuration attributes which are merged and gives an explanation of their behavior.

Configuration Attribute	Merged?	Behavior
Rules (Groups, Users, Scripted, Custom, Device and Process)	Yes	The merged configuration contains all the Rules from each of the separate configurations. If two rules which affect the same application exist in the same trigger, they will run in parallel. The contents of individual rules are not merged.
Application Groups	Yes	The merged configuration will contain all Application Groups from the configuration layers. Application Groups remain unique to the non-base configuration and are all merged into the final merged configuration.
URM Policies	Yes	The merged configuration will contain all URM Policies from the configuration layers. URM Policies remain unique to the non-base configurations and are all merged into the final merged configuration.
Auditing	No	The events from the base are used in the merged configuration whilst the events from the component configurations are ignored.
Engineering Keys	No	Merged configurations inherit their Engineering Keys from the base configuration. Settings from any component configuration in the merge are discarded. It is therefore important that the Engineering Keys you require in the merged configuration are added to the base configuration.
Default Options	No	Merged configurations inherit their Default Options from the base configuration, for example Enabling User Privilege Management. Settings from any component configuration in the merge are discarded. It is therefore important that the Default keys you require in the merged configuration are added to the base configuration.
Message Settings	No	Merged configurations inherit their Settings from the base configuration, for example the message a user will see when a user self-elevates an item. Settings from any component configuration in the merge are discarded.
Archive	No	Merged configurations inherit their Archive Settings from the base

Configuration Attribute	Merged?	Behavior
Settings		configuration, for example Enabling archiving. Settings from any component configuration in the merge are discarded.
Privilege Discovery Mode Configurations	No	Merged configurations inherit their Privilege Discovery Mode Settings from the base configuration, for example hidden applications and files. Settings from any component configuration in the merge are discarded.
URL Redirection Setting	No	Merged configurations inherit any URL Settings from the base configuration, for example configured connection types or IP ranges. Settings from any component configuration in the merge are discarded.
Audit Event Filtering Settings	No	Merged configurations inherit their Audit Event Filtering Settings from the base configuration. Settings from any component configuration in the merge are discarded.

Live Configuration Rules

When a live configuration is opened or saved on an endpoint, it is referred to:

%ProgramData%\AppSense\Application Manager\Configuration\configuration.aamp

To allow for configuration merging, the live configuration can also refer to:

%ProgramData%\AppSense\Application Manager\Configuration\Merged_Configuration.aamp

The agent monitors the %ProgramData%\AppSense\Application Manager\Configuration directory for new configurations. When a change is detected the agent loads a new configuration using the following order of precedence:

1. Merged_Configuration.aamp
2. configuration.aamp

If a Merged_Configuration.aamp exists in the directory, it will be the live configuration. If removed, the agent continues to use the in-memory version - it will not switch to the configuration.aamp file until the agent is restarted.

Live Configuration Update Behavior

The BaseConfigMergeBehavior engineering key allows you to define how the live configuration is affected when a Configuration.aamp file is pushed out to endpoints by the Management Center or other deployment method.

By defining the BaseConfigMergeBehavior engineering key, you can modify the live configuration behavior:

- **Remerge** - When the configuration is detected on an endpoint by the agent, a merge, based on the last_merge_manifest.xml, is triggered and includes the new configuration.aamp. The merge creates a new Merged_Configuration.aamp which replaces the current live configuration. A last_merge_manifest.xml must be present otherwise the merge will fail.
- **Replace** - When the configuration is detected on an endpoint by the agent, it replaces the Merged_Configuration.aamp as the live configuration. Following the successful deployment of the new Configuration.aamp, the Merged_Configuration.aamp is deleted from the directory.

Endpoint Configuration Merging Auditing Events

New auditing events for Configuration Endpoint Merging have been added to Application Control. When viewed in Windows Event Viewer (select **Windows Log** > **Applications**), the events provide further details such as what has caused a merge failure

Endpoint Analysis

Endpoint Analysis (EPA) allows you to scan single or multiple endpoints, to provide a list of applications that are present and that have run on a particular computer. Endpoint Analysis helps to simplify the creation of an appropriate Application Control configuration. This feature is used on demand and is inactive by default. There are two ways to analyze the data on endpoints with the Application Control Agent installed:

- **Endpoint Scans** - Endpoint Analysis files for a given endpoint are stored on the computer that has the Application Control console installed under the following location:

C:\ProgramData\AppSense\Application Manager\EndpointAnalysis.

The Endpoint Scan searches the endpoint for any applications that are present. These applications may have been officially installed by an administrator, or be an esoteric piece of virus-ridden freeware installed by an unsuspecting end user.

The following directory and registry locations are scanned:

- HKLM\SOFTWARE\Microsoft\Windows\Current\CurrentVersion\Installer\Folders
 - HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
 - Program Files
- **Application Usage Scans** - The Application Usage Scan is used to detect all applications in use on an endpoint. When an Application Usage Scan is in progress, all execute requests are passed through for Endpoint Analysis processing once the standard Application Control rules checking has been performed on that request. The details of requests are held in memory. When the scan is stopped, all the request data is saved to file.

If the endpoint is rebooted while a scan is in progress, for example, if a user takes their laptop from the workplace and switches it on at home, the Endpoint Analysis runtime detects that it should be recording application usage and restarts the recording. This is done on agent startup.

An Endpoint Scan can take several minutes. The reason for this is that Application Control not only scans the Program Files folder and the registry keys, but also each dependent file and digital signatures. Application Control records all this information.

During an Endpoint Scan, 100% of the CPU on the endpoint can be used. However, if user tasks need to be performed, the Application Control agent uses built-in smart scheduling technology to allow tasks to take precedence over the scan itself, so the end-user perception of performance is unaffected.

Typically, the Endpoint Scan is run first to determine which applications are installed on the endpoint. This can be followed by the Application Usage Scan to track the applications that have been run on an endpoint over a period of time. By highlighting which applications are being used and which are not, unlicensed software can be identified and restricted and unlicensed software can be removed. Before either scan is run, endpoints must be specified in the Endpoint Analysis tree.

Endpoint Analysis Preparation

For Endpoint Analysis to function the following must be installed:

- Application Control agent installed on the endpoint.
- License installed on the endpoint.
- Application Control configuration installed on the endpoint.
- Administrative share rights to the endpoint.
- Remote registry access to the endpoint.

Test that the Agent is Installed on the Endpoint

1. On the Start menu select **Control Panel**.
2. Select **Administrative Tools**.
3. Double-click **Services**.
4. Locate the Application Control Agent.

Test that the License is Installed on the Endpoint

1. Launch the Registry Editor on the managed endpoint.
2. Locate the license under HKLM\Software\AppSense Technologies\Licensing.

Test that the Configuration is Installed on the Endpoint

Configurations are stored in the following location:

C:\ProgramData\AppSense\ApplicationManager\Configuration.



ProgramData is a hidden folder. Open up explorer and type C:\ProgramData in the Address bar. Press Enter to open the folder.

Test that the Endpoint has Admin Share Rights

1. Open Windows Explorer on the computer that has the Application Control console installed.
2. In the Address bar enter \\<computername>c\$ and press Enter. If you can browse the folders, you have access rights. If not, you are prompted for user credentials that allow access.

Test that Remote Registry Access is Available

1. Open the Registry Editor on the computer that has the Application Control console installed.
2. Select **File > Connect Network Registry**.
3. The Select Computer dialog is displayed.
4. Locate the computer and click **OK**. If you can see the registry keys, you have access.



On remote computers running Windows 7 and above, File Sharing and Remote Registry Service are disabled by default and must be enabled.

5. Turn on File Sharing in **Start > Control Panel > Network and Sharing Center**.
6. Start the Remote Registry Service in **Start > Control Panel > Administrative Tools > Services**.

Working with Endpoint Analysis

This feature provides the ability to perform the endpoint and Application Usage scans and to show all loaded files (child processes) for scanned applications and any digital certificates for the discovered applications.

It is recommended to include all loaded files in the configuration for an Accessible Item so that the application functions correctly. It is also useful to add any digital certificates to the Trusted Vendors in your configuration.

Add Endpoints

Endpoints must be specified before they can be scanned.

1. Click the **Endpoint Analysis** navigation button.
2. The Endpoint Analysis navigation tree displays.
3. From the Endpoint Analysis ribbon, click **Add Endpoint** and select one of the following:
 - **Browse Deployment Group** - The Select Management Server dialog displays. Navigate to the deployment group location and select the required endpoints.
 - **Browse Domain / Workgroup** - The Add Endpoints for Analysis dialog displays. Enter the name or IP address or use the ellipsis (...) in the Computer field to select the required endpoints and click **Add**.

The endpoint displays in the Endpoint Analysis navigation tree. Once added, an endpoint can be used in Endpoint Analysis.

To remove an endpoint, highlight it and click the **Remove Endpoint** button in the Endpoint Analysis ribbon.

Installed Applications Scans

Run scans on selected endpoints within a specified domain. The scan checks the following directories and registry locations:

- HKLM\SOFTWARE\Microsoft\Windows\Current\CurrentVersion\Installer\Folders
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
- Program Files

Once the scan is complete, a report is generated detailing applications and files installed on scanned endpoints. Other information, such as DLLs and digitally signed files that are spawned as a result of running an application executable, are also captured.

Run an Endpoint Scan

Perform an Endpoint Scan on endpoints where the Application Control Agent is installed.

1. Select the **Endpoint Analysis** navigation button.

To run the Endpoint Scan, you must first add endpoints. For information, see [Add Endpoints](#).

2. Select an endpoint and click **Run Endpoint Scan**. To scan all the endpoints within selected domain, click **Run Scan for all Endpoints**.
3. The Endpoint Scan checks the following directories and registry locations:
 - HKLM\SOFTWARE\Microsoft\Windows\Current\CurrentVersion\Installer\Folders
 - HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
 - Program Files

The results of the Endpoint Scan display in the Installed Applications node, nested under the relevant endpoint. Details such as the Application Name, Application Description and Owner are available. For additional file information, use the following Endpoint Analysis ribbon buttons:

- **Show Loaded Files** - Displays details of other files that are loaded by applications.
- **Show Digital Certificates** - Displays details of certificates assigned to applications.

Application Usage Scans

Application Usage Scans detect all applications that are running during the scan period that have not been installed using Windows Installer technology (MSIs and MSPs), such as an executable that runs whilst extracting a ZIP file, in-house software, or Firefox. Start Application Usage scans at any time to monitor actively used applications when users are logged on to an endpoint. Stop the usage scan at any time to generate a report and save it as an XML data file. The data file contains details of applications used on scanned endpoints.

Export the XML data file for archiving purposes or import the file to other endpoints. For example, as data from scans is only available on the endpoint that ran the scan, another administrator can import the exported data file and use the data to create an Application Control configuration.

Alternatively, the data file can be imported into the Rules Analyzer to troubleshoot the behavior of Application Control by using the information contained in the data file.

For information on configuring and analyzing rules, see [Rules Analyzer](#).

Run an Application Usage Scan

Perform an Application Usage Scans on a managed endpoint when a user is logged in.

1. Click the **Endpoint Analysis** navigation button.

The Endpoint Analysis navigation tree displays.



To run the Application Usage Scan, you must first add endpoints. For information, see [Adding Endpoints](#).

2. In the navigation tree, select the endpoint to be scanned.

The Endpoint Summary work area displays.

3. From the Endpoint Analysis ribbon, click **Start Application Usage Scan**.

The Application Scan begins. The scan can be run for however long it takes for you to collect the required data and stopped when enough data has been collected.

4. Click **Stop Application Usage Scan** to stop the scan and generate a report.

The Save Report dialog displays.

5. Enter a name for the report and click **OK** to save the data file.

The file is saved in XML format and created under the Recorded Data node for the selected endpoint.

Application Data

The application data can be seen in detail for both the Installed Applications Scan and the Application Usage Scan.

You can select to display the associated loaded files or the digital certificates:

- **Show Loaded Files** - displays the Loaded files dialog. Drag and drop any of the files to add to the configuration.
- **Show Digital Certificates** - displays the Certificates dialog. Drag and drop any of the certificates to add to any of the **Trusted Vendors** node in the configuration.



On occasion a duplicate certificate will be present, for example: Calc.exe loads Msvcr7.dll, Ntdll.dll and Msutil.dll. Calc.exe is signed with 'Microsoft Certificate A' and Ntdll.dll is also signed with 'Microsoft Certificate A'. Refer to the Signed File column to clearly identify which file has been signed with which certificate.

Export an Endpoint Analysis Data File

Export data files to be imported into other endpoint or the Rules Analyzer.

1. Select the endpoint from which the data file is to be exported.
2. From the Endpoint Analysis ribbon, click **Export**.
The Export browser dialog displays.
3. Select a location to save the file.
4. Click **Save**.

The data file is saved to the selected location and can be imported into other Application Control consoles or the Rules Analyzer.

Add Files to Configurations

Use the results of Endpoint Analysis to add rules, for applications and files, to the Application Control Configuration file. Drag and drop applications, files, DLLs, or certificates into the Group Rules available from the Rules node, accessed from the Configuration navigation button.

If you drag and drop files into any of the Accessible or Prohibited Items lists they are dropped in as files.

- If files are placed in Accessible Items, any associated loaded files are automatically included.
- If files are placed in Prohibited Items, any associated loaded files are not included, only the main application executable.

To add a certificate to any of the Trusted Vendors you can either drag and drop a file on to a Trusted Vendors node (if any certificates exist for that file they are added) or you can select **Show Digital Signatures** on the Endpoint Analysis ribbon to display the Certificates dialog. You can then drag and drop from that dialog into the configuration.



When you drag and drop files into a configuration, the digital signature for the file is always copied over as this is the most secure method for authenticating an application.

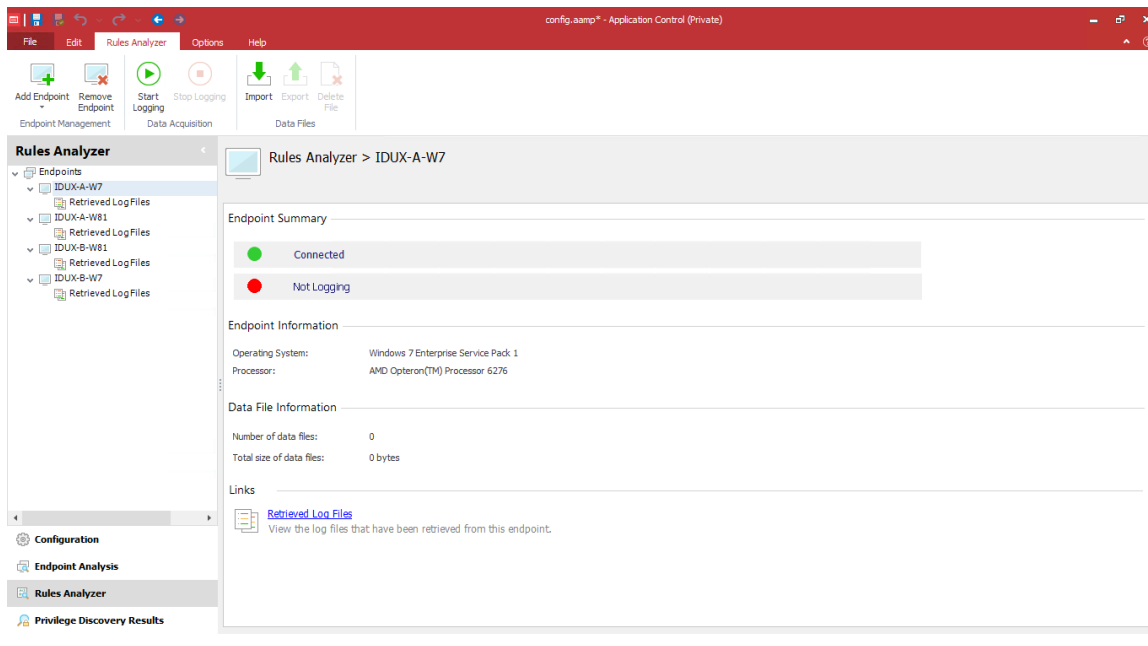
Rules Analyzer

Standard auditing can be used to track unauthorized application usage or to track when users are overwriting \ renaming applications. It is a simple mechanism to use and can function without interaction. The standard auditing mechanism advises you when an application has not, for example, been allowed to execute but does not advise why this was the case. Therefore, an additional tool is required so you can analyze the rules base in real time, and determine exactly why an application is or is not allowed to execute.

Rules Analyzer examines managed endpoints to collect information about how Application Control Rules are applied and provides details of any inconsistencies or inaccuracies in rules as they are processed. Rules Analyzer provides you with a graphical interface that can be used to manually troubleshoot and fine tune Application Control configurations in real time anywhere across the enterprise. All that is required is a network link to a remote Application Control managed endpoint so the Rules Analyzer can connect to the agent software and start logging on the local endpoint.

When the logging has completed you can use the Rules Analyzer to automatically pull the log file across the network back to the computer where the analysis is occurring for investigation. All logging information is held in XML format and each execution request that the Application Control agent processed is listed along with the details of what occurred during processing, including if the process was allowed to execute or not and the reason for the outcome.

The Console



The Rules Analyzer is accessed from the navigation pane within the Application Control console and is used to create, retrieve and examine the log files.

An Endpoint node allows you to control logging on to a specific managed endpoint to retrieve the log files. Below each Endpoint node is a node for each Retrieved Log Files node.

You can review a summary page, view all requests, or view the requests for a specific user. You can restrict the view to the denied or allowed requests. Within the analysis panel you can navigate to a specific request and view the full details of that request, including which rules were applied by Application Control.

You must be logged on with an account that allows read and write access to the registry of any managed endpoint for which you wish to generate logs for using Rules Analyzer, and have read and write access to the local registry of the computer on which the console operates.

Prerequisites

Test that the following are in place:

The Application Control agent is installed on the endpoint.

1. On the Start menu select **Control Panel**.
2. Select **Administrative Tools**.
3. Double-click **Services**.
4. Locate the Application Control Agent.

A license is installed on the endpoint.

1. Launch the Registry Editor on the managed endpoint.
2. Locate the license under HKLM\Software\AppSense Technologies\Licensing.

An Application Control configuration is installed on the endpoint.

1. Configurations are stored in the following location:
2. Navigate to C:\ProgramData\AppSense\ApplicationManager\Configuration.



ProgramData is a hidden folder. Open Windows explorer and type C:\ProgramData in the Address bar. Press Enter to open the folder.

You have Admin share privileges on the endpoint.

1. Open Windows Explorer on the computer that has the Application Control console installed.
2. In the Address bar enter \\<computername>\c\$ and press Enter. If you can browse the folders, you have access rights. If not, you are prompted for user credentials that allow access.

Remote registry access is available on the endpoint

1. Open the Registry Editor on the computer that has the Application Control console installed.
2. Select **File > Connect Network Registry**.
3. The Select Computer dialog is displayed.
4. Locate the computer and click **OK**. If you can see the registry keys, you have access.

On remote computers running Windows 7 and above, File Sharing and Remote Registry Service are disabled by default and must be enabled to ensure Rules Analyzer can access or create log files:

1. Turn on File Sharing in **Start > Control Panel > Network and Sharing Center**.
2. Start the Remote Registry Service in **Start > Control Panel > Administrative Tools > Services**.

Set Up Logging for Rules Analyzer

The first requirement is to add an endpoint to the list of endpoints that the Rules Analyzer can interact with.

Add an Endpoint

Endpoints must be specified before rules are analyzed.

1. Click the **Rules Analyzer** navigation button.
The Rules Analyzer navigation tree displays.
2. From the Rules Analyzer ribbon, click **Add Endpoint** and select one of the following:
 - **Browse Deployment Group** - The Select Management Server dialog displays. Navigate to the deployment group location and select the required endpoints.
 - **Browse Domain / Workgroup** - The Add Rules Analyzer Endpoints dialog displays. Enter the name or IP address or use the ellipsis (...) in the Computer field to select the required endpoints and click **Add**.
3. The endpoint displays in the Rules Analyzer navigation tree. Once added, an endpoint can be analyzed by the Rules Analyzer.
4. To remove an endpoint, highlight it and click the **Remove Endpoint** button in the Rules Analyzer ribbon.

Start and Stop Logging

1. Select the endpoint in the navigation tree.
2. Select **Start Logging** on the Rules Analyzer ribbon.
3. When required, for example, after you have recreated a problem on the endpoint, select **Stop Logging** on the Rules Analyzer ribbon.
The File dialog is displayed.
4. Enter a name for the log file and click **OK**.
5. The XML file is displayed in the navigation tree.



Rules Analyzer files can be large so this feature should only be used when a problem manifests itself and investigation is required.

Once you have created the log files, you can export them or delete them by selecting the files and using the Export and Delete buttons in the Rules Analyzer ribbon.

You can also import log files in XML format by selecting an endpoint and clicking Import in the Rules Analyzer ribbon.

Log Files

All log files for a given computer are stored on the local machine during logging and are temporarily stored in the following location:

C:\Documents and Settings\All Users\Application Data\AppSense\ApplicationManager\Rules Analyzer\RulesAnalyzerLog.xml.

When logging is stopped on the specific endpoint, the log file is closed and transferred to the computer that is running the Rules Analyzer, where it is stored in the cache for the endpoint in question. The cache is held in the following location:

C:\Documents and Settings\All Users\Application Data\AppSense\ApplicationManager\Rules Analyzer\

The naming convention for the files is ComputerName^enteredname. For example, C:\Documents and Settings\All Users\Application Data\AppSense\ApplicationManager\Rules Analyzer\APPUKTECHPUBS2^Regedit.xml.

The computer name is the name of the endpoint as it is entered in the user interface. Therefore, if it is an IP address it is stored as IPAddress^enteredname.xml. The entered name is the name given to the XML file in the Rules Analyzer.

The Rules Analyzer console displays the information regarding execution requests in a number of ways to enable easy access to the details:

Log File Contents Summary

The **Endpoint Summary** page displays when you select a log file node in the navigation tree.

It shows the number of requests processed by Application Control. The top row of the table shows the total number of requests for all users. The remaining rows show the number of requests for each user. The Total column shows the total number of requests, allowed and denied. The Allowed/Denied column shows the number of allowed or denied requests.

Click on any Total link to display the Log File Contents Request List.

To export the log file in XML format, select **Export** on the Rules Analyzer ribbon.



You can select View the requests by processing time on the Summary page to display a Request List page showing requests sorted with the longest running request first.

Log File Contents Request List

The Request List page displays a list of Application Control requests when you click a Total link in the Summary page.

The requests are listed in the order in which they were processed by Application Control.

Each request displays a green tick or red cross indicating to indicate whether the request was allowed or denied.

Click on a request link to display the Log File Contents Request Details.

Log File Contents Request Details

The Request Detail page displays details of a particular request when you click a request in the Request List page.

The Request Detail page displays each rule applied by Application Control in processing the request. The rules are listed in the order applied. The last rule in the list determines the final result – allow or deny. The rule information includes links which, when selected, display popup messages providing explanations explanation for the rule item.



Use the Return link at the top of the page to navigate to the previous page and the Summary link to return to the Summary page. The Back button on the console toolbar is for navigating the navigation tree.

Rules Analyzer Tasks

Common Rules Analyzer tasks include:

- **Analyze a log file** - To analyze a log file, select the log file node. The first page shown in the analysis work area is the summary page. You navigate inside the analysis panel by following links. Use the Return link at the top of the page to go back to the previous page.
- **View requests for a specific user** - To view the requests for a specific user click one of the links in the table on the summary page. You can click in the Total column to see all the requests for the user and you can click in the Allowed column or the Denied column to see only the allowed or denied requests.
- **Find requests that take a long time** - To find requests that take a long time click **View** the requests by processing time on the summary page. This shows the requests sorted, with the longest running request first. The processing time shown is the elapsed time taken by the AppSense Application Control agent to process the request.

Sample Scripting Reference

The following are Visual Basic script examples showing common operations that can be performed with the Application Control scripting interface:

Creating New Configurations

Create a new configuration and save to file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the default configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.DefaultConfiguration
Configuration.ParseXML ConfigurationXml
ConfigurationHelper.SaveLocalConfiguration "C:\Configuration.aamp",Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Create a new configuration and save to live configuration

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the default configuration
Configuration.ParseXML ConfigurationHelper.DefaultConfiguration
'Save the blank configuration to file.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Loading Live Configurations

Load configuration from file and save to file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the Live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
```

```
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the configuration from file
Dim ConfigurationXml
ConfigurationXml =
ConfigurationHelper.LoadLocalConfiguration("C:\Configuration.aamp")
Configuration.ParseXML ConfigurationXml
ConfigurationHelper.SaveLocalConfiguration "C:\Configuration.aamp",
Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```


Default Rules

Edit a default rules configuration

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Configuration.DefaultRules.AllowCMDForBatchFiles = True
Configuration.DefaultRules.ValidateSystemProcesses = False
'Add a trusted owner to the configuration
Dim theTrustedOwner
Set theTrustedOwner = Configuration.CreateInstanceFromClassName("AM.TrustedOwner")
theTrustedOwner.DisplayName = "%COMPUTERNAME%\Guest"
theTrustedOwner.SID = "S-1-5-Domain-501"
Configuration.DefaultRules.TrustedOwners.Add theTrustedOwner.Xml
'Save the configuration to file.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```



The DefaultConfiguration() method only returns a configuration in the English language. This means that some group names and other text in the configuration may not be in the native language of the operating system, which can result in the configuration not being applied correctly. For non-English operating systems it is necessary to export the default configuration from the product console on a native operating system. This can be stored as a file on the network or distributed to the machine where the configuration scripting will be performed. Once this is done, use the LoadLocalConfiguration() method in place of the DefaultConfiguration(). This will produce the same configuration but in the correct native language.

Group Rules

Create a group rule

```
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Dim GroupRule
Set GroupRule = Configuration.CreateInstanceFromClassName("AM.GroupRule")
GroupRule.DisplayName = "BUILTIN\Remote Desktop Users"
GroupRule.Name = GroupRule.DisplayName
GroupRule.SID = "S-1-5-32-555"
Set GroupRule = Configuration.GroupRules.Add(GroupRule.Xml)
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit a group rule

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Change the SID of the Everyone group
Configuration.GroupRules.Item("Everyone").SID = "S-1-1-0"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete a group rule

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Create the group rule
Dim GroupRule
Set GroupRule = Configuration.CreateInstanceFromClassName("AM.GroupRule")
GroupRule.DisplayName = "BUILTIN\Remote Desktop Users"
GroupRule.Name = GroupRule.DisplayName
GroupRule.SID = "S-1-5-32-555"
Configuration.GroupRules.Add GroupRule.Xml
'Delete the rule
Configuration.GroupRules.Remove "BUILTIN\Remote Desktop Users"
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
```

```
Set ConfigurationHelper = Nothing  
Set Configuration = Nothing
```

User Rules

Create a user rule

```
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Create and add the new user rule
Dim UserRule
Set UserRule = Configuration.CreateInstanceFromClassName("AM.UserRule")
UserRule.DisplayName = "%COMPUTERNAME%\Guest"
UserRule.Name = UserRule.DisplayName
UserRule.SID = "S-1-5-Domain-501"
Configuration.UserRules.Add UserRule.Xml
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit a user rule

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Modify the user rule
Dim UserRule
Set UserRule = Configuration.UserRules.Item("%COMPUTERNAME%\Guest")
UserRule.SID = "S-1-5-Domain-501"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete a user rule

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Modify the user rule
Dim UserRule
Set UserRule = Configuration.UserRules.Item("%COMPUTERNAME%\Guest")
UserRule.SID = "S-1-5-Domain-501"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
```

```
Set ConfigurationHelper = Nothing  
Set Configuration = Nothing
```

Device Rules

Create a device rule

```
' Constant definitions for the AM.HostType enumeration.
const AM_DeviceType_Computer = 0
const AM_DeviceType_ConnectingDevice = 1
const AM_DeviceType_Either = 2
' Constant definitions for the AM.HostNameType enumeration.
const AM_HostNameType_HostName = 0
const AM_HostNameType_IPAddress = 1
const AM_HostNameType_ComputerGroup = 2
const AM_HostNameType_OU = 3
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Create a device rule
Dim DeviceRule
Set DeviceRule = Configuration.CreateInstanceFromClassName("AM.DeviceRule")
DeviceRule.Name = "Device Rule (1)"
Configuration.DeviceRules.Add DeviceRule.Xml
'Add a device to the rule
Dim Device
Set Device = Configuration.CreateInstanceFromClassName("AM.Device")
Device.Host = "MyComputer"
Device.NameType = AM_HostNameType_HostName
Configuration.DeviceRules.Item("Device Rule (1)").Devices.Add Device.Xml
'Add another device to the rule
Dim AnotherDevice
Set AnotherDevice = Configuration.CreateInstanceFromClassName("AM.Device")
AnotherDevice.Host = "192.168.0.2"
AnotherDevice.NameType = AM_HostNameType_IPAddress
Configuration.DeviceRules.Item("Device Rule (1)").Devices.Add AnotherDevice.Xml
Configuration.DeviceRules.Item("Device Rule (1)").Devices.Item("192.168.0.2").HostType = AM_DeviceType_ConnectingDevice
'Add device using Computer Group
Dim ComputerGroupMembership
Set ComputerGroupMembership = Configuration.CreateInstanceFromClassName("AM.Device")
ComputerGroupMembership.Host = "CN=Finance,OU=Administration,OU=Corporate,DC=myDomain"
ComputerGroupMembership.NameType = AM_HostNameType_ComputerGroup
Configuration.DeviceRules.Item("Device Rule (1)").Devices.Add ComputerGroupMembership.Xml
'Add device using OU
Dim OUMembership
Set OUMembership = Configuration.CreateInstanceFromClassName("AM.Device")
OUMembership.Host = "OU=HR,OU=Administration,OU=Corporate,DC=myDomain"
OUMembership.NameType = AM_HostNameType_OU
Configuration.DeviceRules.Item("Device Rule (1)").Devices.Add OUMembership.Xml
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit a device rule

```
' Constant definitions for the AM.SecurityLevel enumeration.
const AM_SecurityLevel_Restricted = 0
const AM_SecurityLevel_SelfAuthorizing = 1
const AM_SecurityLevel_Unrestricted = 2
const AM_SecurityLevel_AuditOnly = 3
' Constant definitions for the AM.HostType enumeration.
const AM_DeviceType_Computer = 0
const AM_DeviceType_ConnectingDevice = 1
const AM_DeviceType_Either = 2
' Constant definitions for the AM.HostNameType enumeration.
const AM_HostNameType_HostName = 0
const AM_HostNameType_IPAddress = 1
const AM_HostNameType_ComputerGroup = 2
const AM_HostNameType_OU = 3
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Create a device rule
Dim DeviceRule
Set DeviceRule = Configuration.CreateInstanceFromClassName("AM.DeviceRule")
DeviceRule.Name = "Device Rule (1)"
Configuration.DeviceRules.Add DeviceRule.Xml
Configuration.DeviceRules.Item("Device Rule (1)").Name = "My Device Rule"
Configuration.DeviceRules.Item("Device Rule (1)").SecurityLevel =
AM_SecurityLevel_AuditOnly
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete a device rule

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Remove "Device Rule(1)"
Configuration.DeviceRules.Remove "Device Rule (1)"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Scripted Rules

Create a scripted rule

```
' Constant definitions for the AM.ExecutionContext enumeration.
const AM_ExecutionContext_PerSessionAsUser = 0
const AM_ExecutionContext_PerSessionAsSystem = 1
const AM_ExecutionContext_PerComputerAsSystem = 2
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Create the scripted rule.
Dim ScriptedRule
Set ScriptedRule = Configuration.CreateInstanceFromClassName("AM.ScriptedRule")
ScriptedRule.Name = "Scripted Rule (1)"
Configuration.ScriptedRules.Add ScriptedRule.Xml
Configuration.ScriptedRules.Item("Scripted Rule (1)").WaitForLogin = True
Configuration.ScriptedRules.Item("Scripted Rule (1)").Script = "Function
ScriptedRule()" & Chr(10) & "'Test scripted rule" & Chr(10) & "ScriptedRule=TRUE" &
Chr(10) & "End Function"
Configuration.ScriptedRules.Item("Scripted Rule (1)").EntryFunction = "ScriptedRule"
Configuration.ScriptedRules.Item("Scripted Rule (1)").Timeout = 6
Configuration.ScriptedRules.Item("Scripted Rule (1)").Context =
AM_ExecutionContext_PerSessionAsSystem
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit a scripted rule

```
' Constant definitions for the AM.ExecutionContext enumeration.
const AM_ExecutionContext_PerSessionAsUser = 0
const AM_ExecutionContext_PerSessionAsSystem = 1
const AM_ExecutionContext_PerComputerAsSystem = 2
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Create the scripted rule.
Dim ScriptedRule
Set ScriptedRule = Configuration.CreateInstanceFromClassName("AM.ScriptedRule")
ScriptedRule.Name = "Scripted Rule (1)"
Configuration.ScriptedRules.Add ScriptedRule.Xml
Dim CurrentScriptedRule
For Each CurrentScriptedRule in Configuration.ScriptedRules
If CurrentScriptedRule.Name = "Scripted Rule (1)" Then
CurrentScriptedRule.Timeout = 7
End If
Next
```



```
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete a scripted rule

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Remove the scripted rule.
Configuration.ScriptedRules.Remove "Scripted Rule (1)"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Browser Control

Add URL Redirection Item

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Application Manager configuration files have the aamp file extension. This file contains many
different files which together
'become our Configuration file. One of these is the ConfigurationXml. This is the file where all of
AM's rules are configured.
'However the aamp file contains other files which play a smaller part in the configuration.
'We are now providing a Save/Load routine combination which will allow the user to overwrite the
configurationXml whilst preserving the
'other files unchanged in the aamp file. The normal Load/Save routines would cause a new file to be
created containing only the configurationXml
'Calling the LoadLocalConfigurationHandleWithAuditing routine passes back the configuration xml as
the return value, but also the Auditing xml
' and a FileHandle.
'use this file handle in the equivalent save routine and it will preserve any non-configuration
files - in the aamp file.
Dim FileHandle
Dim ConfigurationXml
Dim AuditingXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfigurationHandleWithAuditing(AuditingXml,
FileHandle)
Configuration.ParseXML ConfigurationXml
' Create a new URL Redirection item
Dim UrlItem
Set UrlItem = Configuration.CreateInstanceFromClassName("AM.URLRedirectionItem")
UrlItem.Path = "bbc.co.uk"
UrlItem.CustomRedirectionUrl = "http://www.appsense.com"
UrlItem.RedirectToCustomUrl = True
UrlItem.UseRegularExpression = False
UrlItem.Description = "Add description here"
' Add the URL Redirection item to the Everyone group
Configuration.GroupRules.Item("Everyone").UrlRedirectionURLs.Add UrlItem.xml
'Saves the ConfigurationXml and Auditing xml to the configuration aamp file whilst preserving any
other existing files contained in it.
ConfigurationHelper.SaveLiveConfigurationHandleWithAuditing Configuration.Xml, AuditingXml,
FileHandle
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Configure Properties

Message settings

```
' Constant definitions for the AM.ANACMessageFrequencyType enumeration.
const AM_ANACMessageFrequencyType_EveryConnectionAttempt = 0
const AM_ANACMessageFrequencyType_Once = 1
const AM_ANACMessageFrequencyType_UseDelayBetweenMessages = 2
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Modify the message settings
Configuration.MessageSettings.AccessDeniedMessageCaption = "Warning"
Configuration.MessageSettings.AccessDeniedMessageBody = "File has been blocked"
Configuration.MessageSettings.ApplicationLimitsExceededMessageCaption = "Warning"
Configuration.MessageSettings.ApplicationLimitsExceededMessageBody = "Too many files"
Configuration.MessageSettings.DisplayInitialWarningMessage = False
Configuration.MessageSettings.CloseApplication = False
Configuration.MessageSettings.TerminateApplication = False
Configuration.MessageSettings.WaitTime = 120
Configuration.MessageSettings.TimeLimitsWarningMessageCaption = "Warning"
Configuration.MessageSettings.TimeLimitsWarningMessageBody = "Out of time"
Configuration.MessageSettings.TimeLimitsDeniedMessageCaption = "Warning"
Configuration.MessageSettings.TimeLimitsDeniedMessageBody = "Wrong time"
Configuration.MessageSettings.SelfAuthorizationMessageCaption = "Warning"
Configuration.MessageSettings.SelfAuthorizationMessageBody = "Needs authorization"
Configuration.MessageSettings.SelfAuthorizationResponseCaption = "Authorized File"
Configuration.MessageSettings.SelfAuthorizationResponseBody = "File is now authorized."
Configuration.MessageSettings.ANACMessageBoxEnabled = True
Configuration.MessageSettings.ANACMessageFrequency =
AM_ANACMessageFrequencyType_Once
Configuration.MessageSettings.ANACMessageDelayBetweenMessageBoxes = 60
Configuration.MessageSettings.ANACMessageBoxCaption = "Application Manager -
Application Network Access Control"
Configuration.MessageSettings.ANACMessageBoxBody = "%ExecutableName% has been denied access to
%NetworkLocation%."
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Archive options

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Modify the archiving settings
Dim ArchiveFolder
Set ArchiveFolder = Configuration.CreateInstanceFromClassName("AM.ArchiveFolder")
```

```
ArchiveFolder.Path = "C:\ArchiveBackup"
Set ArchiveFolder =
Configuration.ArchivingSettings.ArchiveFolders.InsertBefore(ArchiveFolder.Xml, 1)
Configuration.ArchivingSettings.ArchivingEnabled = True
Configuration.ArchivingSettings.AnonymousEnabled = True
Configuration.ArchivingSettings.UserLimit = 26
Configuration.ArchivingSettings.TotalLimit = 51
Configuration.ArchivingSettings.NoAdminOwnedFiles = True
Configuration.ArchivingSettings.OverwriteExistingFiles = False
Configuration.ArchivingSettings.ArchiveLessThanEnabled = True
Configuration.ArchivingSettings.OverwriteOldest = True
Configuration.ArchivingSettings.ArchiveLessThanAmount = 10
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Application termination

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Modify the Application Termination Settings
Configuration.ApplicationTerminationSettings.Enabled = True
Configuration.ApplicationTerminationSettings.DisplayInitialWarningMessage = True
Configuration.ApplicationTerminationSettings.CloseApplication = True
Configuration.ApplicationTerminationSettings.TerminateApplication = True
Configuration.ApplicationTerminationSettings.WaitTime = 60
'Modify the Application Termination Triggers
Configuration.ApplicationTerminationSettings.Triggers.TerminateOnConfigurationChange = True
Configuration.ApplicationTerminationSettings.Triggers.TerminateOnComputerIPAddressChanged = False
Configuration.ApplicationTerminationSettings.Triggers.TerminateOnConnectingDeviceChanged = True
' Modify the Application Termination Messages
Configuration.MessageSettings.ApplicationTerminationMessages.ConfigAppliedWarningMessageCaption =
"New Configuration Applied Message Caption"
Configuration.MessageSettings.ApplicationTerminationMessages.ConfigAppliedWarningMessageBody = "New
Configuration Applied Message Body"
' The other Termination Message objects are:
'
' ConfigAppliedTerminateMessageCaption
' ConfigAppliedTerminateMessageBody
' IPAddressChangedWarningMessageCaption
' IPAddressChangedWarningMessageBody
' IPAddressChangedTerminateMessageCaption
' IPAddressChangedTerminateMessageBody
' ConnectingDeviceChangedWarningMessageCaption
' ConnectingDeviceChangedWarningMessageBody
' ConnectingDeviceChangedTerminateMessageCaption
' ConnectingDeviceChangedTerminateMessageBody
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Add Engineering Key

```
'Create the configuration
```

```
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Add an Engineering key
Dim EngineeringKey
Set EngineeringKey = Configuration.CreateInstanceFromClassName("AM.EngineeringKey")
EngineeringKey.Name = "UrmSecPolicy"
EngineeringKey.Value = "1"
Configuration.EngineeringKeys.Add EngineeringKey.Xml
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Network Connections

Add network connection

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Add a connection to the list of accessible connections.
Dim AccessibleConn
Set AccessibleConn =
Configuration.CreateInstanceFromClassName("AM.NetworkConnection")
AccessibleConn.Path = "www.google.com:80/foo/*"
AccessibleConn.Address = "www.google.com"
AccessibleConn.Port = 80
AccessibleConn.Resource = "/foo/*"
AccessibleConn.UseWildcards = True
AccessibleConn.AddressType = 0
Configuration.GroupRules.Item("Everyone").AccessibleNetworkConnections.Add
AccessibleConn.Xml
'Add a connection to the list of prohibited connections.
Dim ProhibitedConn
Set ProhibitedConn =
Configuration.CreateInstanceFromClassName("AM.NetworkConnection")
ProhibitedConn.Path = "www.facebook.com"
ProhibitedConn.AddressType = 0
ProhibitedConn.Description = "www.facebook.com"
Configuration.GroupRules.Item("Everyone").ProhibitedNetworkConnections.Add
ProhibitedConn.Xml
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit a network connection

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Modify the port number of the network connection
Configuration.GroupRules.Item("Everyone").AccessibleNetworkConnections.Item("www.google.com:80/foo/*").Port = 8080
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete a network connection

```
'Create the configuration
```

```
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Remove network conenction
Configuration.GroupRules.Item("Everyone").ProhibitedNetworkConnections.Remove
"www.facebook.com"
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Process Rules

Create a process rule

```
Create a Process Rule
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Create a process rule
Dim ProcessRule
Set ProcessRule = Configuration.CreateInstanceFromClassName("AM.ProcessRule")
ProcessRule.Name = "Process Rule (1)"
Configuration.ProcessRules.Add ProcessRule.Xml
'Add a file process to the rule
Dim FileProcess
Set FileProcess = Configuration.CreateInstanceFromClassName("AM.File")
FileProcess.Path = "c:\windows\system32\notepad.exe"
FileProcess.CommandLine = "c:\windows\system32\notepad.exe"
Configuration.ProcessRules.Item("Process Rule (1)").FileProcessItems.AddFileProcess.Xml
'Add another file to the rule
Dim AnotherFile
Set AnotherFile = Configuration.CreateInstanceFromClassName("AM.File")
AnotherFile.Path = "c:\windows\system32\cmd.exe"
AnotherFile.CommandLine = "c:\windows\system32\cmd.exe"
Configuration.ProcessRules.Item("Process Rule (1)").FileProcessItems.AddAnotherFile.Xml
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit a process rule

```
' Constant definitions for the AM.SecurityLevel enumeration.
const AM_SecurityLevel_Restricted = 0
const AM_SecurityLevel_SelfAuthorizing = 1
const AM_SecurityLevel_Unrestricted = 2
const AM_SecurityLevel_AuditOnly = 3
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Configuration.ProcessRules.Item("Process Rule (1)").Name = "My Process Rule"
Configuration.ProcessRules.Item("My Process Rule").SecurityLevel =
AM_SecurityLevel_AuditOnly
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```


Delete a process rule

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Remove "Process Rule(1)"
Configuration.ProcessRules.Remove "Process Rule (1)"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Rules List Items

Add a file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Add a file to the list of accessible files.
Dim AccessibleFile
Set AccessibleFile = Configuration.CreateInstanceFromClassName("AM.File")
AccessibleFile.Path = "calc.exe"
AccessibleFile.Commandline = "calc.exe"
Configuration.GroupRules.Item("Everyone").AccessibleFiles.Add AccessibleFile.Xml
'Add a file to the list of prohibited files.
Dim ProhibitedFile
Set ProhibitedFile = Configuration.CreateInstanceFromClassName("AM.File")
ProhibitedFile.Path = "regedit.exe"
ProhibitedFile.CommandLine = "regedit.exe"
Configuration.GroupRules.Item("Everyone").ProhibitedFiles.Add ProhibitedFile.Xml
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit a file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Edit calc.exe.
Configuration.GroupRules.Item("Everyone").AccessibleFiles.Item("calc.exe").TrustedOwnershipChecking = False
Configuration.GroupRules.Item("Everyone").AccessibleFiles.Item("calc.exe").ApplicationLimit = 5
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete a file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
```

```

ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Remove files
Configuration.GroupRules.Item("Everyone").AccessibleFiles.Remove "calc.exe"
Configuration.GroupRules.Item("Everyone").ProhibitedFiles.Remove "regedit.exe"
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing

```

Add a folder

```

'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Dim AccessibleFolder
Set AccessibleFolder = Configuration.CreateInstanceFromClassName("AM.Folder")
AccessibleFolder.Path = "%ALLUSERSPROFILE%"
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Add AccessibleFolder.Xml
Dim ProhibitedFolder
Set ProhibitedFolder = Configuration.CreateInstanceFromClassName("AM.Folder")
ProhibitedFolder.Path = "%SystemDrive%\Utilities"
Configuration.GroupRules.Item("Everyone").ProhibitedFolders.Add ProhibitedFolder.Xml
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing

```

Edit a folder

```

'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.Recursive = False
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.AccessTimes.MondayTimeRangeCollection.Clear()
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.AccessTimes.TuesdayTimeRangeCollection.Clear()
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.AccessTimes.WednesdayTimeRangeCollection.Clear()
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.AccessTimes.ThursdayTimeRangeCollection.Clear()
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.AccessTimes.FridayTimeRangeCollection.Clear()
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.AccessTimes.SaturdayTimeRangeCollection.Clear()
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.AccessTimes.SundayTimeRangeCollection.Clear()
Dim TimeRange

```

```
Set TimeRange = Configuration.CreateInstanceFromClassName("AM.TimeRange")
TimeRange.StartHour = 9
TimeRange.EndHour = 13
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.AccessTimes.MondayTimeRangeCollection.InsertBefore TimeRange.Xml, 0
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Item("%ALLUSERSPROFILE%")
.ApplyAccessTimes = True
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete a folder

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Remove the accessible folder
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Remove
"%ALLUSERSPROFILE%"
'Remove the prohibited folder
Configuration.GroupRules.Item("Everyone").ProhibitedFolders.Remove
"%SystemDrive%\Utilities"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Add a digital signature

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Create new signature item
Dim SignatureFile
Set SignatureFile = Configuration.CreateInstanceFromClassName("AM.SignatureFile")
SignatureFile.SHA1Hash =
ConfigurationHelper.ReadShalHashFromFile("C:\WINDOWS\regedit.exe")
SignatureFile.Path = "C:\WINDOWS\regedit.exe"
SignatureFile.CommandLine = SignatureFile.SHA1Hash
'Add the signature to the rule
Configuration.GroupRules.Item("Everyone").AccessibleSignatures.Add SignatureFile.Xml
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit a digital signature

```
'Create the configuration
```

```
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Digital signatures are keyed by CommandLine, containing the SHA1 hash, so obtain
the
hash value to access the required item.
Dim sha1Hash
sha1Hash = ConfigurationHelper.ReadSha1HashFromFile("C:\WINDOWS\regedit.exe")
Configuration.GroupRules.Item("Everyone").AccessibleSignatures.Item(sha1Hash).ApplyAccessTimes
= False
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete a digital signature

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Digital signatures are keyed by SHA1 hash, so obtain the hash value to access the
required item.
Dim sha1Hash
sha1Hash = ConfigurationHelper.ReadSha1HashFromFile("C:\WINDOWS\regedit.exe")
Configuration.GroupRules.Item("Everyone").AccessibleSignatures.Remove sha1Hash
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Add and delete drives

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Add first drive
Dim FirstDrive
Set FirstDrive = Configuration.CreateInstanceFromClassName("AM.Drive")
FirstDrive.Path = "H"
Configuration.GroupRules.Item("Everyone").AccessibleDrives.Add FirstDrive.Xml
'Add a second drive
Dim SecondDrive
Set SecondDrive = Configuration.CreateInstanceFromClassName("AM.Drive")
SecondDrive.Path = "I"
```

```
Configuration.GroupRules.Item("Everyone").AccessibleDrives.Add SecondDrive.Xml
'Remove the first drive that was added
Configuration.GroupRules.Item("Everyone").AccessibleDrives.Remove "H"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Add a trusted vendor

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Use the helper object to read the certificate from the signed file
Dim CertificateData
CertificateData = ConfigurationHelper.ReadCertificateFromFile("C:\Program
Files\Internet Explorer\iexplore.exe", 0)
Dim DigitalCertificate
Set DigitalCertificate =
Configuration.CreateInstanceFromClassName("AM.DigitalCertificate")
DigitalCertificate.RawCertificateData = CertificateData
DigitalCertificate.Description = "Microsoft Corporation - Internet Explorer
Certificate"
Set DigitalCertificate =
Configuration.GroupRules.Item("Everyone").TrustedVendors.Add(DigitalCertificate.Xml)
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Add a trusted vendor and their certificate expiry date

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Use the helper object to read the certificate and expiry date from the signed file
Dim CertificateData
Dim dtMyDate
CertificateData = ConfigurationHelper.ReadCertificateDateFromFile("C:\Program
Files\Internet Explorer\iexplore.exe", 0, dtMyDate)
'Add the certificate information to the configuration
Dim DigitalCertificate
Set DigitalCertificate =
Configuration.CreateInstanceFromClassName("AM.DigitalCertificate")
DigitalCertificate.RawCertificateData = CertificateData
DigitalCertificate.Description = "Microsoft Corporation - Internet Explorer
Certificate"
DigitalCertificate.ExpiryDate = dtMyDate
Set DigitalCertificate =
Configuration.GroupRules.Item("Everyone").TrustedVendors.Add(DigitalCertificate.Xml)
```

```
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit a trusted vendor

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Use the helper object to read the certificate from the signed file
Dim CertificateData
CertificateData = ConfigurationHelper.ReadCertificateFromFile("C:\Program
Files\Internet Explorer\iexplore.exe", 0)
Configuration.GroupRules.Item("Everyone").TrustedVendors.Item(CertificateData).Enfor
ceExpiryDate = True
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete a trusted vendor

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Use the helper object to read the certificate from the signed file
Dim CertificateData
CertificateData = ConfigurationHelper.ReadCertificateFromFile("C:\Program
Files\Internet Explorer\iexplore.exe", 0)
Configuration.GroupRules.Item("Everyone").TrustedVendors.Remove CertificateData
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Group Management

Add Library groups

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Create a new Group in the Library
Dim LibraryGroup
Set LibraryGroup = Configuration.CreateInstanceFromClassName("AM.ApplicationGroup")
LibraryGroup.Path = "Common Applications"
Dim CommonFile
Set CommonFile = Configuration.CreateInstanceFromClassName("AM.File")
CommonFile.Path = "calc.exe"
CommonFile.Commandline = "calc.exe"
LibraryGroup.Files.Add CommonFile.Xml
Dim NotepadFile
Set NotepadFile = Configuration.CreateInstanceFromClassName("AM.File")
NotepadFile.Path = "notepad.exe"
NotepadFile.Commandline = "notepad.exe"
LibraryGroup.Files.Add NotepadFile.Xml
Configuration.ApplicationGroups.Add LibraryGroup.Xml
'Save the configuration to file.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Use Library groups

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Use an existing Library Group in the Rules
Dim GroupReference
Set GroupReference =
Configuration.CreateInstanceFromClassName("AM.ApplicationGroupReference")
GroupReference.Group = "Common Applications"
GroupReference.TrustedOwnershipChecking = "True"
GroupReference.Path = "Common Applications"
Configuration.GroupRules.Item("Everyone").ProhibitedApplicationGroups.Add
GroupReference.Xml
'Save the configuration to file.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```


User Privileges Management

Edit UPM policies

```
'URM Group Action options
const AM_URMGroupAction_Add = 0
const AM_URMGroupAction_Drop = 1
'URM Privilege actions
const AM_URMPrivilegeAction_NoChange = 0
const AM_URMPrivilegeAction_Enable = 1
const AM_URMPrivilegeAction_Disable = 2
const AM_URMPrivilegeAction_Remove = 3
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Configuration.URMPolicies("Add
Administrator").PrivilegeActions("SeBackupPrivilege").Action =
AM_URMPrivilegeAction_Enable
Configuration.URMPolicies("Add
Administrator").GroupMembershipActions("BUILTIN\Administrators").Action =
AM_URMGroupAction_Drop
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete UPM policies

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Configuration.URMPolicies.Remove "Add Administrator"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
Add a user privileges file
Edit a user privileges file
Delete a user privileges file
```

Delete user privileges component

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
```

```
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMWellKnownControlPanelApplets.Remove
"cplClock"
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Auditing

Save to file with auditing file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the default configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.DefaultConfiguration
Configuration.ParseXML ConfigurationXml
Dim AuditingFile
AuditingFile = "c:\Auditing.xml"
ConfigurationHelper.SaveLocalConfigurationWithAuditingFile
"C:\Configuration.aamp", Configuration.Xml, AuditingFile
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Save to live with auditing file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the default configuration
Configuration.ParseXML ConfigurationHelper.DefaultConfiguration
'Save the blank configuration to file.
Dim AuditingFile
AuditingFile = "c:\Auditing.xml"
ConfigurationHelper.SaveLiveConfigurationWithAuditingFile
Configuration.Xml, AuditingFile
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Load file with auditing

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
Dim AuditingXml
ConfigurationXml =
ConfigurationHelper.LoadLiveConfigurationWithAuditing(AuditingXml)
Configuration.ParseXML ConfigurationXml
'Edit some settings
Configuration.DefaultRules.AllowCMDForBatchFiles = False
Configuration.DefaultRules.ValidateSystemProcesses = False
'Save the configuration to file.
ConfigurationHelper.SaveLiveConfigurationWithAuditing Configuration.Xml, AuditingXml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Load live with auditing

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
Dim AuditingXml
ConfigurationXml =
ConfigurationHelper.LoadLocalConfigurationWithAuditing("c:\Configuration.aamp", AuditingXml)
Configuration.ParseXML ConfigurationXml
'Edit settings
Configuration.DefaultRules.AllowCMDForBatchFiles = False
Configuration.DefaultRules.ValidateSystemProcesses = False
'Save the configuration to file.
ConfigurationHelper.SaveLocalConfigurationWithAuditing
"C:\UpdatedConfiguration.aamp", Configuration.Xml, AuditingXml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Additional Load and Save functions

Load local configuration handle with auditing

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
Dim AuditingXml
Dim FileHandle
'Application Managers configuration files have the aamp file extension. This file contains many
different files which together
'become our Configuration file. One of these is the ConfigurationXml. This is the file where all of
AM's rules are configured.
'However the aamp file contains other files which play a smaller part in the configuration.
'We are now providing a Save/Load routine combination which will allow the user to overwrite the
configurationXml whilst preserving the
'other files unchanged in the aamp file. The normal Load/Save routines would cause a new file to be
created containing only the configurationXml
'Calling the LoadLocalConfigurationHandleWithAuditing routine passes back the configuration xml as
the return value, but also the Auditing xml
' and a FileHandle.
'use this file handle in the equivalent save routine and it will preserve any nonconfiguration files
- in the aamp file.
ConfigurationXml =
ConfigurationHelper.LoadLocalConfigurationHandleWithAuditing("c:\temp\configuration.aamp",
AuditingXml, FileHandle)
Configuration.ParseXML ConfigurationXml
Configuration.DefaultRules.AllowCMDForBatchFiles = True
Configuration.DefaultRules.ValidateSystemProcesses = True
'Saves the ConfigurationXml and Auditing xml to the configuration aamp file whilst preserving any
other existing files contained in it.
ConfigurationHelper.SaveLocalConfigurationHandleWithAuditing
"c:\temp\configuration.aamp", Configuration.Xml, AuditingXml, FileHandle
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Load live configuration handle with auditing

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
Dim AuditingXml
Dim FileHandle
'Application Managers configuration files have the aamp file extension. This file contains many
different files which together
'become our Configuration file. One of these is the ConfigurationXml. This is the file where all of
AM's rules are configured.
'However the aamp file contains other files which play a smaller part in the configuration.
'We are now providing a Save/Load routine combination which will allow the user to overwrite the
configurationXml whilst preserving the
'other files unchanged in the aamp file. The normal Load/Save routines would cause a new file to be
created containing only the configurationXml
```

```
'Calling the LoadLiveConfigurationHandleWithAuditing routine passes back the live configuration xml
as the return value, but also the Auditing xml
' and a FileHandle.
' use this file handle in the equivalent save routine and it will preserve any nonconfiguration files
- in the aamp file.
ConfigurationXml = ConfigurationHelper.LoadLiveConfigurationHandleWithAuditing(
AuditingXml, FileHandle)
Configuration.ParseXML ConfigurationXml
Configuration.DefaultRules.AllowCMDForBatchFiles = True
Configuration.DefaultRules.ValidateSystemProcesses = True
'Saves the ConfigurationXml and Auditing xml to the live configuration whilst preserving any other
existing files.
ConfigurationHelper.SaveLiveConfigurationHandleWithAuditing Configuration.Xml,
AuditingXml, FileHandle
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Load local configuration handle without auditing

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
Dim AuditingXml
Dim FileHandle
'Application Managers configuration files have the aamp file extension. This file contains many
different files which together
'become our Configuration file. One of these is the ConfigurationXml. This is the file where all of
AM's rules are configured.
'However the aamp file contains other files which play a smaller part in the configuration.
'We are now providing a Save/Load routine combination which will allow the user to overwrite the
configurationXml whilst preserving the
'other files unchanged in the aamp file. The normal Load/Save routines would cause a new file to be
created containing only the configurationXml
'Calling the LoadLocalConfigurationHandle routine passes back the configuration xml as the return
value, but also a FileHandle.
' use this file handle in the equivalent save routine and it will preserve any nonconfiguration
files - in the aamp file.
ConfigurationXml =
ConfigurationHelper.LoadLocalConfigurationHandle("c:\temp\configuration.aamp",
FileHandle)
Configuration.ParseXML ConfigurationXml
Configuration.DefaultRules.AllowCMDForBatchFiles = True
Configuration.DefaultRules.ValidateSystemProcesses = True
'Saves the ConfigurationXml to the specified configuration whilst preserving any other existing
files from the FileHandle.
ConfigurationHelper.SaveLocalConfigurationHandle "c:\temp\configuration.aamp",
Configuration.Xml, FileHandle
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Load live configuration handle without auditing

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
```

```
'Load the live configuration
Dim ConfigurationXml
Dim AuditingXml
Dim FileHandle
'Application Manangers configuration files have the aamp file extension. This file contains many
different files which together
'become our Configuration file. One of these is the ConfigurationXml. This is the file where all of
AM's rules are configured.
'However the aamp file contains other files which play a smaller part in the configuration.
'We are now providing a Save/Load routine combination which will allow the user to overwrite the
configurationXml whilst preserving the
'other files unchanged in the aamp file. The normal Load/Save routines would cause a new file to be
created containing only the configurationXml
'Calling the LoadLiveConfigurationHandle routine passes back the live configuration xml as the
return value, but also a FileHandle.
'use this file handle in the equivalent save routine and it will preserve any nonconfiguration files
- in the aamp file.
ConfigurationXml = ConfigurationHelper.LoadLiveConfigurationHandle(FileHandle)
Configuration.ParseXML ConfigurationXml
Configuration.DefaultRules.AllowCMDForBatchFiles = True
Configuration.DefaultRules.ValidateSystemProcesses = True
'Saves the ConfigurationXml to the live configuration whilst preserving any other existing files.
ConfigurationHelper.SaveLiveConfigurationHandle Configuration.Xml, FileHandle
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Adding files and folders with metadata

Add files and folders with metadata

```
'Create a File Item with Metadata which will be used later in the script
'
Set FileWithMetadata = Configuration.CreateInstanceFromClassName("AM.File")
'Set the actual file
FileWithMetadata.Path = "MetadataFile.exe"
'Set a unique key for this item in the collection it is added to
FileWithMetadata.Commandline = "MetadataFile.exe"
'Set some metadata properties
FileWithMetadata.Metadata.ProductVersionMinimum = "*. *.*.*"
FileWithMetadata.Metadata.ProductVersionMinimumEnabled = True
FileWithMetadata.Metadata.ProductVersionMaximum = "*. *.*.*"
FileWithMetadata.Metadata.ProductVersionMaximumEnabled = True
FileWithMetadata.Metadata.FileVersionMinimum = "*. *.*.*"
FileWithMetadata.Metadata.FileVersionMinimumEnabled = True
FileWithMetadata.Metadata.FileVersionMaximum = "*. *.*.*"
FileWithMetadata.Metadata.FileVersionMaximumEnabled = True
FileWithMetadata.Metadata.VendorName = "VEND"
FileWithMetadata.Metadata.VendorNameEnabled = True
FileWithMetadata.Metadata.ProductName = "PROD"
FileWithMetadata.Metadata.ProductNameEnabled = True
FileWithMetadata.Metadata.CompanyName = "COMP"
FileWithMetadata.Metadata.CompanyNameEnabled = True
FileWithMetadata.Metadata.FileDescription = "DESC"
FileWithMetadata.Metadata.FileDescriptionEnabled = True
'Create a Folder Item with Metadata which will be used later in the script
'
Set FolderWithMetadata = Configuration.CreateInstanceFromClassName("AM.Folder")
'Set a unique key for this item in the collection it is added to
FolderWithMetadata.ItemKey = "c:\MetadataFolder"
'Set the actual folder
FolderWithMetadata.Path = "c:\MetadataFolder"
'Set some metadata properties
FolderWithMetadata.Metadata.VendorName = "VEND"
FolderWithMetadata.Metadata.VendorNameEnabled = True
'Add a file to the list of accessible files.
Configuration.GroupRules.Item("Everyone").AccessibleFiles.Add FileWithMetadata.Xml
'Add the file item to a URM Rule
'
'Create the URM Item
Set URMFile = Configuration.CreateInstanceFromClassName("AM.URMRuleItemPolicy")
'Configure the URM Item with the details of the Accessible File created earlier
URMFile.KeyPath = FileWithMetadata.Commandline
URMFile.Application = FileWithMetadata.Xml
'Set the URM Policy to Apply
URMFile.Policy.Policy = "Add Administrator"
'Add the URM item
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMFiles.Add URMFile.xml
'Add a folder to the list of accessible folders.
Configuration.GroupRules.Item("Everyone").AccessibleFolders.Add FolderWithMetadata.Xml
'Add the folder item to a URM Rule
'
'Create the URM Item
Set URMFolder = Configuration.CreateInstanceFromClassName("AM.URMRuleItemPolicy")
'Configure the URM Item with the details of the Accessible Folder created earlier
URMFolder.KeyPath = FolderWithMetadata.ItemKey
URMFolder.Application = FolderWithMetadata.Xml
'Set the URM Policy to Apply
URMFolder.Policy.Policy = "Add Administrator"
'Add the URM item
```



```
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMFolders.Add URMFolder.xml
'Process Rule with Metadata Configuration on the specific process
'
'Create a new Process Rule
Set ProcessRule = Configuration.CreateInstanceFromClassName("AM.ProcessRule")
ProcessRule.Name = "Process Rule With Metadata"
'Add a file process to the rule
ProcessRule.FileProcessItems.Add FileWithMetadata.Xml
'Add the process rule
Configuration.ProcessRules.Add ProcessRule.Xml
'Save the live configuration.
'
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Windows Store Apps

Add Windows Store app

```

const AM_VersionMatching_andabove = 0
const AM_VersionMatching_andbelow = 1
const AM_VersionMatching_exactly = 2
const AM_VersionMatching_allversions = 3
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Prohibit all windows store apps
Dim BlockAllApps
Set BlockAllApps = Configuration.CreateInstanceFromClassName("AM.WindowsStoreApp")
BlockAllApps.DisplayName = "* * All installed apps * *"
BlockAllApps.PackageName = "*"
BlockAllApps.PublisherID = "*"
BlockAllApps.Publisher = "*"
BlockAllApps.PackageVersion = "1.0.0.0"
BlockAllApps.VersionMatch = AM_VersionMatching_allversions
BlockAllApps.Path = "* * 1.0.0.0"
Configuration.GroupRules.Item("Everyone").ProhibitedWindowsStoreApps.Add
BlockAllApps.Xml
'Add a Windows Store App to the list of accessible connections.
Dim AccessibleApp
Set AccessibleApp = Configuration.CreateInstanceFromClassName("AM.WindowsStoreApp")
AccessibleApp.DisplayName = "Skype"
AccessibleApp.PackageName = "Microsoft.SkypeApp"
AccessibleApp.PublisherID = "kzf8qxf38zg5c"
AccessibleApp.Publisher = "CN=Skype Software Sarl, O=Microsoft Corporation,
L=Luxembourg, S=Luxembourg, C=LU"
AccessibleApp.PackageVersion = "3.1.0.1007"
AccessibleApp.VersionMatch = AM_VersionMatching_andabove
AccessibleApp.Path = "kzf8qxf38zg5c_Microsoft.SkypeApp_3.1.0.1007"
Configuration.GroupRules.Item("Everyone").AccessibleWindowsStoreApps.Add AccessibleApp.Xml
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing

```

Edit Windows Store app

```

const AM_VersionMatching_andabove = 0
const AM_VersionMatching_andbelow = 1
const AM_VersionMatching_exactly = 2
const AM_VersionMatching_allversions = 3
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml

```

```
'Modify the Version Matching
Configuration.GroupRules.Item("Everyone").AccessibleWindowsStoreApps.Item("kzf8qxf38zg5c_
Microsoft.SkypeApp_3.1.0.1007").VersionMatch
= AM_VersionMatching_allversions
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete Windows Store app

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Remove Skype
Configuration.GroupRules.Item("Everyone").AccessibleWindowsStoreApps.Remove "kzf8qxf38zg5c_
Microsoft.SkypeApp_3.1.0.1007"
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

System Controls

Adding system controls

```
'BuiltinActions
const AM_ControlPanelURMPolicy_BuiltinElevate = 0
const AM_ControlPanelURMPolicy_BuiltinRestrict = 1
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
Dim AuditingXml
Dim FileHandle
ConfigurationXml = ConfigurationHelper.LoadLiveConfigurationHandle(FileHandle)
Configuration.ParseXML ConfigurationXml
Dim UninstallItem
Set UninstallItem = Configuration.CreateInstanceFromClassName("AM.UninstallControl")
UninstallItem.Path = "AppSense Application Manager"
UninstallItem.DisplayName = "AppSense Application Manager"
UninstallItem.Publisher = "AppSense"
UninstallItem.Version = "8.9.*"
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMUninstallControls.Add
UninstallItem.Xml
Dim EventlogItem
Set EventlogItem = Configuration.CreateInstanceFromClassName("AM.EventlogControl")
EventlogItem.Path = "Application"
EventlogItem.LogName = "Application"
EventlogItem.Policy = AM_ControlPanelURMPolicy_BuiltinElevate
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMEventlogControls.Add
EventlogItem.Xml
Dim ServiceItem
Set ServiceItem = Configuration.CreateInstanceFromClassName("AM.ServiceControl")
ServiceItem.Path = "AppSense Application Manager Agent"
ServiceItem.ServiceDisplayName = "AppSense Application Manager Agent"
ServiceItem.ServiceName = ""
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMServiceControls.Add
ServiceItem.Xml
'Save the configuration to file.
ConfigurationHelper.SaveLiveConfigurationHandle Configuration.Xml, FileHandle
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Self-Elevation Files

Adding self-elevation files

```
'Constant definitions for the AM.SelfElevationFilterMode enumeration.
const AM_SelfElevationFilterMode_AllowAllExcept = 0
const AM_SelfElevationFilterMode_DenyAllExcept = 1
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the default configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.DefaultConfiguration
Configuration.ParseXML ConfigurationXml
'Create a file to add to Self-Elevation files for the "Everyone" group
Dim SelfElevationFile
Set SelfElevationFile = Configuration.CreateInstanceFromClassName("AM.File")
SelfElevationFile.Path = "calc.exe"
SelfElevationFile.Commandline = "calc.exe"
'Add a file to the Self-Elevation tab under User Rights, for the "Everyone" group
Configuration.GroupRules.Item("Everyone").SelfElevationRules.SelfElevationFiles.Add
SelfElevationFile.Xml
'Enable Self-Elevation
Configuration.GroupRules.Item("Everyone").SelfElevationRules.SelfElevationEnabled
= true
'Set Self-Elevation to only apply to items in the list
Configuration.GroupRules.Item("Everyone").SelfElevationRules.Filtermode = AM_
SelfElevationFilterMode_DenyAllExcept
'Make files Accessible Items
Configuration.GroupRules.Item("Everyone").SelfElevationRules.MakeAccessible = true
'Apply User Rights for child processes
Configuration.GroupRules.Item("Everyone").SelfElevationRules.ApplyToChildProcesses
= true
'Allow file to run even if not owned by a Trusted Owner
Configuration.GroupRules.Item("Everyone").SelfElevationRules.TrustedOwnershipChecking=
true
'Apply to Common Dialogs
Configuration.GroupRules.Item("Everyone").SelfElevationRules.ApplyToOpenSave = true
'Install as Trusted Owner
Configuration.GroupRules.Item("Everyone").SelfElevationRules.ChangeOwnershipToAdmin
= true
'Save Configuration to disk
ConfigurationHelper.SaveLocalConfiguration "C:\Configuration.aamp",Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Policy Change Requests

Configuring policy change request settings

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Modify the Demand Config Change Settings
'Enable the global feature
Configuration.OnDemandConfigChangeSettings.OnDemandEnabled = True
'Enable the Email Requests
Configuration.OnDemandConfigChangeSettings.EmailRequestsEnabled = True
Configuration.OnDemandConfigChangeSettings.MailToAddress = "sample@company.com"
'Enable the Emergency Requests
Configuration.OnDemandConfigChangeSettings.EmergencyRequestsEnabled = True
Configuration.OnDemandConfigChangeSettings.HelpDeskPhoneNumber = "0800 900 9000"
Dim key
key = ConfigurationHelper.EncryptSharedKey("hello chris")
Configuration.OnDemandConfigChangeSettings.SharedKey = key
'Configure a link from the AMMessage
Configuration.OnDemandConfigChangeSettings.RequestMethods.AllowLinkFromAMDenied =
True
Configuration.OnDemandConfigChangeSettings.RequestMethods.AMDeniedLinkText = "Click
here to submit a change request"
'Configure a Shell context menu
Configuration.OnDemandConfigChangeSettings.RequestMethods.ShowShellMenu = True
Configuration.OnDemandConfigChangeSettings.RequestMethods.ShellMenuText = "Submit
a change request"
'Configure the desktop link
Configuration.OnDemandConfigChangeSettings.RequestMethods.ShowDesktopIcon = True
Configuration.OnDemandConfigChangeSettings.RequestMethods.DesktopIconText = "Request
Policy Change"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

MSIs in the COMConfigurationHelper

Open MSIs

```
Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadMsiConfiguration("C:\msi\AM8.6.msi")
Configuration.ParseXML ConfigurationXml
'Wscript.Echo ConfigurationXml
'Save the blank configuration to file.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
```

Save MSIs

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'Save the blank configuration to file.
ConfigurationHelper.SaveMsiConfiguration "C:\msi\AMout.msi", ConfigurationXml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

User Rights Management

Add user rights file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'create a new FileItem
Dim File
Set File = Configuration.CreateInstanceFromClassName("AM.File")
File.Path = "notepad.exe"
File.CommandLine = "notepad.exe"
Dim URMItem
Set URMFile = Configuration.CreateInstanceFromClassName("AM.URMRuleItemPolicy")
URMFile.KeyPath = "notepad.exe"
URMFile.Policy.Policy = Configuration.URMPolicies.Item("Add Administrator").Name
URMFile.Application = File.Xml
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMFiles.Add URMFile.xml
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Edit user rights file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'create a new FileItem
Dim File
Set File = Configuration.CreateInstanceFromClassName("AM.File")
File.Path = "notepad.exe"
File.Arguments = "test.txt"
File.CommandLine = "notepad.exe test.txt"
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMFiles.Item("notepad.exe").Application =
File.Xml
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMFiles.Item("notepad.exe").KeyPath =
File.CommandLine
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Delete user rights file

```
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
```



```
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMFiles.Remove "notepad.exe test.txt"
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Add user rights file using built-in elevate

```
'URM BuiltinElevate Policy
const BuiltinElevate_Policy = "516A5D5B-685C-49C3-A4FC-3A54BF6CC392\BUILTINADMIN"
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
'Dim ConfigurationXml
'ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
'Configuration.ParseXML ConfigurationXml
'Load the default configuration
Configuration.ParseXML ConfigurationHelper.DefaultConfiguration
'create a new FileItem
Dim File
Set File = Configuration.CreateInstanceFromClassName("AM.File")
File.Path = "notepad.exe"
File.CommandLine = "notepad.exe"
Dim URMItem
Set URMFile = Configuration.CreateInstanceFromClassName("AM.URMRuleItemPolicy")
URMFile.KeyPath = "notepad.exe"
URMFile.Policy.Policy = BuiltinElevate_Policy
URMFile.Application = File.Xml
'Please Note that ApplyToOpenSave is incorrectly named - the meaning has been "flipped"
' ApplyToOpenSave = False => Apply to Common Dialogs
' ApplyToOpenSave = True => Do NOT apply to Common Dialogs.
URMFile.ApplyToOpenSave = False
URMFile.ApplyToChildProcesses = True
URMFile.ChangeOwnershipToAdmin = True
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMFiles.Add URMFile.xml
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Sample Script: Create UPM Policies

The following VB script creates a user privileges management policy:

```
'URM Group Action options
const AM_URMGroupAction_Add = 0
const AM_URMGroupAction_Drop = 1
'URM Privileges
const AM_URMPrivilegeConstant_SeAssignPrimaryTokenPrivilege = 0
const AM_URMPrivilegeConstant_SeAuditPrivilege = 1
const AM_URMPrivilegeConstant_SeBackupPrivilege = 2
const AM_URMPrivilegeConstant_SeChangeNotifyPrivilege = 3
const AM_URMPrivilegeConstant_SeCreateGlobalPrivilege = 4
```

```

const AM_URMPrivilegeConstant_SeCreatePagefilePrivilege = 5
const AM_URMPrivilegeConstant_SeCreatePermanentPrivilege = 6
const AM_URMPrivilegeConstant_SeCreateSymbolicLinkPrivilege = 7
const AM_URMPrivilegeConstant_SeCreateTokenPrivilege = 8
const AM_URMPrivilegeConstant_SeDebugPrivilege = 9
const AM_URMPrivilegeConstant_SeEnableDelegationPrivilege = 10
const AM_URMPrivilegeConstant_SeImpersonatePrivilege = 11
const AM_URMPrivilegeConstant_SeIncreaseBasePriorityPrivilege = 12
const AM_URMPrivilegeConstant_SeIncreaseQuotaPrivilege = 13
const AM_URMPrivilegeConstant_SeIncreaseWorkingSetPrivilege = 14
const AM_URMPrivilegeConstant_SeLoadDriverPrivilege = 15
const AM_URMPrivilegeConstant_SeLockMemoryPrivilege = 16
const AM_URMPrivilegeConstant_SeMachineAccountPrivilege = 17
const AM_URMPrivilegeConstant_SeManageVolumePrivilege = 18
const AM_URMPrivilegeConstant_SeProfileSingleProcessPrivilege = 19
const AM_URMPrivilegeConstant_SeRelabelPrivilege = 20
const AM_URMPrivilegeConstant_SeRemoteShutdownPrivilege = 21
const AM_URMPrivilegeConstant_SeRestorePrivilege = 22
const AM_URMPrivilegeConstant_SeSecurityPrivilege = 23
const AM_URMPrivilegeConstant_SeShutdownPrivilege = 24
const AM_URMPrivilegeConstant_SeSyncAgentPrivilege = 25
const AM_URMPrivilegeConstant_SeSystemEnvironmentPrivilege = 26
const AM_URMPrivilegeConstant_SeSystemProfilePrivilege = 27
const AM_URMPrivilegeConstant_SeSystemtimePrivilege = 28
const AM_URMPrivilegeConstant_SeTakeOwnershipPrivilege = 29
const AM_URMPrivilegeConstant_SeTcbPrivilege = 30
const AM_URMPrivilegeConstant_SeTimeZonePrivilege = 31
const AM_URMPrivilegeConstant_SeTrustedCredManAccessPrivilege = 32
const AM_URMPrivilegeConstant_SeUndockPrivilege = 33
const AM_URMPrivilegeConstant_SeUnsolicitedInputPrivilege = 34
'URM Privilege actions
const AM_URMPrivilegeAction_NoChange = 0
const AM_URMPrivilegeAction_Enable = 1
const AM_URMPrivilegeAction_Disable = 2
const AM_URMPrivilegeAction_Remove = 3
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
'create a new URMPolicy
Dim URMPolicy
Set URMPolicy = Configuration.CreateInstanceFromClassName("AM.URMPolicy")
URMPolicy.Name = "Add Administrator"
Configuration.URMPolicies.Add URMPolicy.Xml
'Add a Group Behaviour Action
Dim URMBehaviour
Set URMBehaviour = Configuration.CreateInstanceFromClassName("AM.URMGroupBehaviour")
URMBehaviour.DisplayName = "BUILTIN\Administrators"
URMBehaviour.SID = "S-1-5-Domain-544"
URMBehaviour.Action = AM_URMGroupAction_Add
Configuration.URMPolicies("Add Administrator").GroupMembershipActions.Add
URMBehaviour.Xml
'Set up the privilege actions
Dim PrivilegeAction
Set PrivilegeAction = Configuration.CreateInstanceFromClassName("AM.URMPrivilege")
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeAssignPrimaryTokenPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeAssignPrimaryTokenPrivilege

```

```
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeAuditPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeAuditPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeBackupPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeBackupPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeChangeNotifyPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeChangeNotifyPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeCreateGlobalPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeCreateGlobalPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeCreatePagefilePrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeCreatePagefilePrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeCreatePermanentPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeCreatePermanentPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeCreateSymbolicLinkPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeCreateSymbolicLinkPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeCreateTokenPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeCreateTokenPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeDebugPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeDebugPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeEnableDelegationPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeEnableDelegationPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeImpersonatePrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeImpersonatePrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeIncreaseBasePriorityPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeIncreaseBasePriorityPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeIncreaseQuotaPrivilege"
```

```
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeIncreaseQuotaPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeIncreaseWorkingSetPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeIncreaseWorkingSetPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeLoadDriverPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeLoadDriverPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeLockMemoryPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeLockMemoryPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeMachineAccountPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeMachineAccountPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeManageVolumePrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeManageVolumePrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeProfileSingleProcessPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeProfileSingleProcessPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeRelabelPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeRelabelPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeRemoteShutdownPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeRemoteShutdownPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeRestorePrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeRestorePrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeSecurityPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeSecurityPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeShutdownPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeShutdownPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeSyncAgentPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeSyncAgentPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
```

```

PrivilegeAction.Name = "SeSystemEnvironmentPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeSystemEnvironmentPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeSystemProfilePrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeSystemProfilePrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeSystemtimePrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeSystemtimePrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeTakeOwnershipPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeTakeOwnershipPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeTcbPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeTcbPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeTimeZonePrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeTimeZonePrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeTrustedCredManAccessPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeTrustedCredManAccessPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeUndockPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeUndockPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
PrivilegeAction.Action = AM_URMPrivilegeAction_NoChange
PrivilegeAction.Name = "SeUnsolicitedInputPrivilege"
PrivilegeAction.Privilege = AM_URMPrivilegeConstant_SeUnsolicitedInputPrivilege
Configuration.URMPolicies("Add Administrator").PrivilegeActions.Add
PrivilegeAction.Xml
'Save the live configuration
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing

```

Sample Script: Add User Privileges Component

The following VB Script adds a user privileges component:

```

const AM_URMControlPanelConstant_mmcComputerManagement = 0
const AM_URMControlPanelConstant_cplAddHardware = 1
const AM_URMControlPanelConstant_cplAddRemovePrograms = 2
const AM_URMControlPanelConstant_cplAddPlugPlay = 3
const AM_URMControlPanelConstant_cplAutomaticUpdatesSettings = 4
const AM_URMControlPanelConstant_cplClock = 5
const AM_URMControlPanelConstant_cplDesktopDPI = 6
const AM_URMControlPanelConstant_cplDisplay = 7
const AM_URMControlPanelConstant_cplInternetOptions = 8
const AM_URMControlPanelConstant_cplPowerOptions = 9

```

```
const AM_URMControlPanelConstant_cplRegional = 10
const AM_URMControlPanelConstant_cplSystem = 11
const AM_URMControlPanelConstant_cplFirewallSettings = 12
const AM_URMControlPanelConstant_mmcFirewallAdvanced = 13
const AM_URMControlPanelConstant_mmcDeviceManager = 14
const AM_URMControlPanelConstant_mmcDiskManagement = 15
const AM_URMControlPanelConstant_cplIndexingOptions = 16
const AM_URMControlPanelConstant_cplWindowsFeatures = 17
const AM_URMControlPanelConstant_mmcLocalSecurityPolicy = 18
const AM_URMControlPanelConstant_mmcPerformanceMonitor = 19
const AM_URMControlPanelConstant_cplLanguages = 20
const AM_URMControlPanelConstant_mmcServices = 21
const AM_URMControlPanelConstant_mmcDefrag = 27
const AM_URMControlPanelConstant_cplBackupRestore = 28
const AM_URMControlPanelConstant_cplScsiInitiator = 29
const AM_URMControlPanelConstant_cplOfflineFiles = 30
const AM_URMControlPanelConstant_cpladaptors = 31
const AM_URMControlPanelConstant_cplprinters = 32
const AM_URMControlPanelConstant_mmcServerManager = 33
const AM_URMControlPanelConstant_cplSystemConfig = 34
const AM_URMControlPanelConstant_cplClearTypeText = 35
const AM_URMControlPanelConstant_cplCalibrateColor = 36
const AM_URMControlPanelConstant_mmcCompServices = 37
const AM_URMControlPanelConstant_cplRecoveryDisc = 38
const AM_URMControlPanelConstant_mmcCertManager = 39
const AM_URMControlPanelConstant_cplDataSources = 40
const AM_URMControlPanelConstant_cplRecoveryRestore = 41
const AM_URMControlPanelConstant_mmcTasksSchedule = 42
const AM_URMControlPanelConstant_mmcTrustedPlatform = 43
const AM_URMControlPanelConstant_cplTroubleShoot = 44
const AM_URMControlPanelConstant_cplBitLockerEnable = 45
const AM_URMControlPanelConstant_mmcEventViewer = 46
const AM_URMControlPanelConstant_cplEasyTransfer = 47
const AM_URMControlPanelConstant_cpladaptorsAdvancedSharing = 48
const AM_URMControlPanelConstant_cpladaptorsWirelessProfile = 49
const AM_URMControlPanelConstant_cpladaptorsWirelessPropertiesChars = 50
const AM_URMControlPanelConstant_cpladaptorsWirelessPropertiesCopyUSB = 51
const AM_URMControlPanelConstant_cpladaptorsNetworkConnectionProperties = 52
const AM_URMControlPanelConstant_cpladaptorsNetworkDisableConnection = 53
const AM_URMControlPanelConstant_cplFirewallSettingsControlPanel = 54
const AM_URMControlPanelConstant_cplFirewallSettingsActionCenter = 55
const AM_URMControlPanelConstant_cplProblemReporting = 56
const AM_URMControlPanelConstant_cplAddRemoveProgramsChange = 57
const AM_URMControlPanelConstant_cplAddRemoveProgramsUninstallUpdate = 58
const AM_URMControlPanelConstant_cplWindowsDefender = 59
const AM_URMControlPanelConstant_cplDefaultLocation = 60
const AM_URMControlPanelConstant_cplAccessCenter = 61
const AM_URMControlPanelConstant_cplExplorer = 62
const AM_URMControlPanelConstant_cplExplorerCheckDisk = 63
const AM_URMControlPanelConstant_cplExplorerEditGroupUser = 64
const AM_URMControlPanelConstant_cplExplorerPermissions = 65
const AM_URMControlPanelConstant_cplExplorerQuota = 66
const AM_URMControlPanelConstant_cplExplorerAdvancedSharing = 67
const AM_URMControlPanelConstant_cplIndexingOptionsAdvanced = 68
const AM_URMControlPanelConstant_cplIndexingOptionsShowAllLocations = 69
const AM_URMControlPanelConstant_cplIndexingOptionsPause = 70
const AM_URMControlPanelConstant_cplMediaSharing = 71
const AM_URMControlPanelConstant_cplUserAccounts = 72
const AM_URMControlPanelConstant_cplUserAccountsUserAccountControl = 73
const AM_URMControlPanelConstant_cplUserAccountsManageUserAccounts = 74
'BuiltinActions
const AM_ControlPanelURMPolicy_BuiltinElevate = 0
const AM_ControlPanelURMPolicy_BuiltinRestrict = 1
'Create the configuration
```

```

Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Dim Applet
Set Applet =
Configuration.CreateInstanceFromClassName("AM.WellKnownControlPanelApplet")
Applet.Path = "cplClock"
Applet.ControlPanelId = AM_URMControlPanelConstant_cplClock
Applet.PolicyAction = AM_ControlPanelURMPolicy_BuiltinElevate
Dim DateTimeComponent
Set DateTimeComponent = Configuration.CreateInstanceFromClassName("AM.URMRuleItem")
DateTimeComponent.KeyPath = Applet.Path
DateTimeComponent.Application = Applet.Xml
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMWellKnownControlPanelApplets.Add DateTimeComponent.Xml
Applet.Path = "mmcServices"
Applet.ControlPanelId = AM_URMControlPanelConstant_mmcServices
Applet.PolicyAction = AM_ControlPanelURMPolicy_BuiltinElevate
DateTimeComponent.KeyPath = Applet.Path
DateTimeComponent.Application = Applet.Xml
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMWellKnownControlPanelApplets.Add DateTimeComponent.Xml
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing

```

Sample Script: Edit User Privileges Component

The following VB script edits a user privileges component:

```

const AM_URMControlPanelConstant_mmcComputerManagement = 0
const AM_URMControlPanelConstant_cplAddHardware = 1
const AM_URMControlPanelConstant_cplAddRemovePrograms = 2
const AM_URMControlPanelConstant_cplAddPlugPlay = 3
const AM_URMControlPanelConstant_cplAutomaticUpdatesSettings = 4
const AM_URMControlPanelConstant_cplClock = 5
const AM_URMControlPanelConstant_cplDesktopDPI = 6
const AM_URMControlPanelConstant_cplDisplay = 7
const AM_URMControlPanelConstant_cplInternetOptions = 8
const AM_URMControlPanelConstant_cplPowerOptions = 9
const AM_URMControlPanelConstant_cplRegional = 10
const AM_URMControlPanelConstant_cplSystem = 11
const AM_URMControlPanelConstant_cplFirewallSettings = 12
const AM_URMControlPanelConstant_mmcFirewallAdvanced = 13
const AM_URMControlPanelConstant_mmcDeviceManager = 14
const AM_URMControlPanelConstant_mmcDiskManagement = 15
const AM_URMControlPanelConstant_cplIndexingOptions = 16
const AM_URMControlPanelConstant_cplWindowsFeatures = 17
const AM_URMControlPanelConstant_mmcLocalSecurityPolicy = 18
const AM_URMControlPanelConstant_mmcPerformanceMonitor = 19
const AM_URMControlPanelConstant_cplLanguages = 20
const AM_URMControlPanelConstant_mmcServices = 21
const AM_URMControlPanelConstant_mmcDefrag = 27
const AM_URMControlPanelConstant_cplBackupRestore = 28
const AM_URMControlPanelConstant_cplScsiInitiator = 29
const AM_URMControlPanelConstant_cplOfflineFiles = 30

```

```

const AM_URMControlPanelConstant_cpladaptors = 31
const AM_URMControlPanelConstant_cplprinters = 32
const AM_URMControlPanelConstant_mmcServerManager = 33
const AM_URMControlPanelConstant_cplSystemConfig = 34
const AM_URMControlPanelConstant_cplClearTypeText = 35
const AM_URMControlPanelConstant_cplCalibrateColor = 36
const AM_URMControlPanelConstant_mmcCompServices = 37
const AM_URMControlPanelConstant_cplRecoveryDisc = 38
const AM_URMControlPanelConstant_mmcCertManager = 39
const AM_URMControlPanelConstant_cplDataSources = 40
const AM_URMControlPanelConstant_cplRecoveryRestore = 41
const AM_URMControlPanelConstant_mmcTasksSchedule = 42
const AM_URMControlPanelConstant_mmcTrustedPlatform = 43
const AM_URMControlPanelConstant_cplTroubleShoot = 44
const AM_URMControlPanelConstant_cplBitLockerEnable = 45
const AM_URMControlPanelConstant_mmcEventViewer = 46
const AM_URMControlPanelConstant_cplEasyTransfer = 47
const AM_URMControlPanelConstant_cpladaptorsAdvancedSharing = 48
const AM_URMControlPanelConstant_cpladaptorsWirelessProfile = 49
const AM_URMControlPanelConstant_cpladaptorsWirelessPropertiesChars = 50
const AM_URMControlPanelConstant_cpladaptorsWirelessPropertiesCopyUSB = 51
const AM_URMControlPanelConstant_cpladaptorsNetworkConnectionProperties = 52
const AM_URMControlPanelConstant_cpladaptorsNetworkDisableConnection = 53
const AM_URMControlPanelConstant_cplFirewallSettingsControlPanel = 54
const AM_URMControlPanelConstant_cplFirewallSettingsActionCenter = 55
const AM_URMControlPanelConstant_cplProblemReporting = 56
const AM_URMControlPanelConstant_cplAddRemoveProgramsChange = 57
const AM_URMControlPanelConstant_cplAddRemoveProgramsUninstallUpdate = 58
const AM_URMControlPanelConstant_cplWindowsDefender = 59
const AM_URMControlPanelConstant_cplDefaultLocation = 60
const AM_URMControlPanelConstant_cplAccessCenter = 61
const AM_URMControlPanelConstant_cplExplorer = 62
const AM_URMControlPanelConstant_cplExplorerCheckDisk = 63
const AM_URMControlPanelConstant_cplExplorerEditGroupUser = 64
const AM_URMControlPanelConstant_cplExplorerPermissions = 65
const AM_URMControlPanelConstant_cplExplorerQuota = 66
const AM_URMControlPanelConstant_cplExplorerAdvancedSharing = 67
const AM_URMControlPanelConstant_cplIndexingOptionsAdvanced = 68
const AM_URMControlPanelConstant_cplIndexingOptionsShowAllLocations = 69
const AM_URMControlPanelConstant_cplIndexingOptionsPause = 70
const AM_URMControlPanelConstant_cplMediaSharing = 71
const AM_URMControlPanelConstant_cplUserAccounts = 72
const AM_URMControlPanelConstant_cplUserAccountsUserAccountControl = 73
const AM_URMControlPanelConstant_cplUserAccountsManageUserAccounts = 74
'BuiltinActions
const AM_ControlPanelURMPolicy_BuiltinElevate = 0
const AM_ControlPanelURMPolicy_BuiltinRestrict = 1
'Create the configuration
Dim Configuration
Set Configuration = CreateObject("AM.Configuration.5")
'Create the configuration helper
Dim ConfigurationHelper
Set ConfigurationHelper = CreateObject("AM.ConfigurationHelper.1")
'Load the live configuration
Dim ConfigurationXml
ConfigurationXml = ConfigurationHelper.LoadLiveConfiguration
Configuration.ParseXML ConfigurationXml
Dim Applet
Set Applet =
Configuration.CreateInstanceFromClassName("AM.WellKnownControlPanelApplet")
Applet.Path = "cplClock"
Applet.ControlPanelId = AM_URMControlPanelConstant_cplClock
Applet.PolicyAction = AM_ControlPanelURMPolicy_BuiltinRestrict
Configuration.GroupRules.Item("Everyone").UserRightsRules.URMWellKnownControlPanelAp

```



```
plets.Item("cplClock").Application = Applet.Xml
'Save the live configuration.
ConfigurationHelper.SaveLiveConfiguration Configuration.Xml
Set ConfigurationHelper = Nothing
Set Configuration = Nothing
```

Configuration Object

The Application Control Object Types include the Configuration object and the Configuration Helper object. The Configuration object represents the Application Control configuration. It is solely concentrated on data and contains no business logic.

Generic Base Types for Collections

Map

Methods:

Add(ValueType item)

Description: Adds a new item into the collection.

Parameters: item - The value to be added.

Remove(KeyType kt)

Description: Removes the value with the given key from the collection.

Parameters: kt - The key of the value to remove from the collection.

Item(KeyType kt)

Description: Accessor for a value within the collection

Returns: The item (value) with the given key.

Parameters: kt - The key of the requested value.

Array

Methods:

Add(ValueType item)

Description: Adds a new item into the collection.

Parameters: item - the value to be added.

Remove(LONG index)

Description: Removes the item at the given position within the collection.

Parameters: index - The 0-based index of the value to remove.

Item(LONG index)

Description: Accessor for the item (value) at the given position within the collection.

Parameters: index - The 0-based index of the requested value.

Strongly Typed Collections

Collection: ArchiveFolderCollection

BaseType: Array

ValueType: ArchiveFolder

Collection: AuditEventFilterDictionary

BaseType: Map

ValueType: AuditEventFilter

Key: File

Collection: ApplicationGroupDictionary

BaseType: Map

ValueType: ApplicationGroup

Key: Path

Collection: CustomRuleDictionary

BaseType: Map

ValueType: CustomRule

Key: Name

Collection: DeviceDictionary

BaseType: Map

ValueType: Device

Key: Host

Collection: DeviceRuleDictionary

BaseType: Map

ValueType: DeviceRule

Key: Name

Collection: DriveCollection

BaseType: Map

ValueType: Drive

Key: Path

Collection: EngineeringKeyCollection

BaseType: Array

ValueType: EngineeringKey

Collection: FileCollection

BaseType: Map

ValueType: File

Key: CommandLine

Collection: FileExtensionDictionary

BaseType: Map

ValueType: FileExtension

Key: Name

Collection: FolderCollection

BaseType: Map

ValueType: Folder

Key: Path

Collection: GroupRuleDictionary

BaseType: Map

ValueType: GroupRule

Key: DisplayName

Collection: NetworkConnectionCollection

Base Type: Map

Value Type: NetworkConnection

Key: Path

Collection: ProcessRuleDictionary

Base Type: Map

Value Type: ProcessRule

Key: Name

Collection: ScriptedRuleDictionary

BaseType: Map

ValueType: ScriptedRule

Key: Name

Collection: SignatureFileCollection

BaseType: Map

ValueType: SignatureFile

Key: CommandLine

Collection: TimeRangeCollection

BaseType: Array

ValueType: TimeRange

Collection: TrustedApplicationCollection

BaseType: Array

ValueType: TrustedApplication

Collection: TrustedOwnerDictionary

BaseType: Map

ValueType: TrustedOwner

Key: DisplayName

Collection: UserRuleDictionary

BaseType: Map

ValueType: UserRule

Key: DisplayName

Collection: URMPolicyDictionary

BaseType: Map

ValueType: URMPolicy

Key: Name

Collection: URMGroupBehaviourDictionary

BaseType: Map

ValueType: URMGroupBehaviour

Key: DisplayName

Collection: URMPrivilegeDictionary

BaseType: Map

ValueType: URMPrivilege

Key: Name

Collection: URMRuleItemDictionary

BaseType: Map

ValueType: URMRuleItem

Key: KeyPath

Collection: URMRuleItemPolicyDirectory

BaseType: Map

ValueType: URMRuleItemPolicy

Key: KeyPath

Object Definitions**Object: Access Times**

Property	Type	Description
MondayTimeRangeCollection	TimeRangeCollection	A collection of time ranges that are applied on Mondays.
TuesdayTimeRangeCollection	TimeRangeCollection	A collection of time ranges that are applied on Tuesdays.
WednesdayTimeRangeCollection	TimeRangeCollection	A collection of time ranges that are applied on Wednesdays.
ThursdayTimeRangeCollection	TimeRangeCollection	A collection of time ranges that are applied on Thursdays.
FridayTimeRangeCollection	TimeRangeCollection	A collection of time ranges that are applied on Fridays.

Property	Type	Description
SaturdayTimeRangeCollection	TimeRangeCollection	A collection of time ranges that are applied on Saturdays.
SundayTimeRangeCollection	TimeRangeCollection	A collection of time ranges that are applied on Sundays.

Object: ApplicationGroup

Property	Type	Description
Path	BSTR	The name of the Application Group.
Description	BSTR	The description of the group.
Files	FileCollection	Collection of files contained in this group.
Folders	FolderCollection	Collection of folders contained in this group.
SignatureFiles	SignatureFileCollection	Collection of signature files contained in this group.
NetworkConnections	NetworkConnectionCollection	Collection of network connections contained within this group.
Drives	DriveCollection	Collection of drives contained within this group.

Object: ArchiveFolder

Property	Type	Description
Path	BSTR	Full path to folder.

Object: ArchivingSettings

Description	Type	Description
ArchivingEnabled	VARIANT_BOOL	Specify whether to use archiving. Default = False
NoAdminOwnedFiles	VARIANT_BOOL	Enable administrator-owned files to be

Description	Type	Description
		ignored. Default = False
OverwriteExistingFiles	VARIANT_BOOL	Specify whether files copied to the archive should overwrite existing files. Default = True
AnonymousEnabled	VARIANT_BOOL	Specify whether file should have any user information stripped.
TotalLimit	LONG	The maximum size of the archive in MB. Default = 50.
UserLimit	LONG	The maximum size of a user's archive in MB. Default = 25.
ArchiveLessThanEnabled	VARIANT_BOOL	Specify whether only files smaller than a certain size will be archived. Default = False.
ArchiveLessThanAmount	LONG	The maximum size of a file that will be copied to the archive. Default = False
OverwriteOldest	VARIANT_BOOL	Specify whether the oldest file in the archive are overwritten when the archive is full. Default = False.
ArchiveFolders	ArchiveFolderCollection	A list of archive folder locations, the first location in the list will be given preference, the last location given the lowest preference.

Object: AuditEventFilter

Property	Type	Description
File	BSTR	The file name/extension to which this filter will be applied.
Events	BSTR	A semi-colon delimited list of events. For example, 9005;9006;9007.

Object: AuditEventFiltering

Property	Type	Description
Enabled	VARIANT_BOOL	Specify whether event filtering is enabled. Default = True.
Files	AuditEventFilterDictionary	The list of event filters.

Object: Configuration

Description	Type	Description
Info	ConfigurationInfo	Configuration metadata
DefaultRules	DefaultRules	Default rules settings.
MessageSettings	MessageSettings	Settings to allow customization of Application Control generated message boxes.
ArchivingSettings	ArchivingSettings	Options for files that are archived.
UserRules	UserRuleDictionary	Collection of configured user rules.
ApplicationGroups	ApplicationGroupDictionary	Library of Application Groups.
ProcessRules	ProcessRuleDictionary	Collection of configured Process Rules
GroupRules	GroupRuleDictionary	Collection of configured group rules.
DeviceRules	DeviceRuleDictionary	Collection of configured device rules.
CustomRules	CustomRuleDictionary	Collection of configured custom rules.
ScriptedRules	ScriptedRuleDictionary	Collection of configured scripted rules.
EngineeringKeys	EngineeringKeyCollection	Collection of engineering keys.
URMPolicies	URMPolicyDictionary	Library of User rights policies.
AuditEventFilteringSettings	AuditEventFiltering	Options relating to which audit events are reported.

Description	Type	Description
OnDemandConfigChangeSettings	OnDemandConfigChangeSettings	Options relating to Policy Change Requests

Object: ConfigurationInfo

Property	Type	Description
Name B	STR	The name of the configuration.
UniqueIdentifier	BSTR	The unique ID for the configuration.
Version	LONG	The configuration version.
Notes	BSTR	Any appropriate notes.
RevisionLevel	LONG	The configuration

Object: CustomRule

Description	Type	Description
DisplayName	BSTR	The account name.
SID	BSTR	The account SID.
Devices	DeviceDictionary	Collection of devices to which this rule applies.
Name	BSTR	The name of the rule.
SecurityLevel	SecurityLevel	The level of restriction applied to this rule.
AccessibleApplicationGroups	ApplicationGroupReferenceDictionary	Collection of allowed Application Groups.
AccessibleFiles	FileCollection	Collection of allowed files
AccessibleFolders	FolderCollection	Collection of allowed folders.
AccessibleDrives	DriveCollection	Collection of allowed drives.
AccessibleSignatures	SignatureFileCollection	Collection of allowed

Description	Type	Description
		signatures.
AccessibleNetworkConnections	NetworkConnectionCollection	Collection of accessible network connections.
ProhibitedApplicationGroups	ApplicationGroupReferenceDictionary	Collection of denied Application Groups.
ProhibitedFiles	FileCollection	Collection of denied files.
ProhibitedFolders	FolderCollection	Collection of denied folders.
ProhibitedDrives	DriveCollection	Collection of denied drives.
ProhibitedSignatures	SignatureFileCollection	Collection of denied signatures.
ProhibitedNetworkConnections	NetworkConnectionCollection	Collection of denied network connections.
TrustedVendors	DigitalCertificateCollection	Collection of trusted vendors' digital certificates.
UserRightsRules	URMRules	Configured settings for user privileges rules.

Object: DefaultRules

Description	Type	Description
TrustedOwnershipChecking	VARIANT_BOOL	Enable trusted ownership checking. Default = True
ChangeFileOwnershipOnOverwriteOrRename	VARIANT_BOOL	Enable a change of file ownership when a file is overwritten or renamed. Default = True

Description	Type	Description
TrustedOwners	TrustedOwnerDictionary	A collection of configured Trusted Owners.
LocalDrivesAccessible	VARIANT_BOOL	Specify whether the local drives are allowed by default. Default = True
IgnoreRestrictionsDuringLogon .	VARIANT_BOOL	Allows restrictions to be ignored until the logon process is complete
AllowCMDForBatchFiles	VARIANT_BOOL	Allows cmd.exe to run if it is run via execution of a batch file. Default = True
ExtractSelfExtractingZIPFiles	VARIANT_BOOL	Specify whether Application Control should extract self-extracting .ZIP files. Default = True
ValidateSystemProcesses	VARIANT_BOOL	Specify whether system process will be subject to Application Control rules processing. Default = False
ValidateMSI	VARIANT_BOOL	Specify whether Windows Installer (.MSI) packages are validated.
ValidateWSH	VARIANT_BOOL	Specify whether Windows Script Host (.WSH) files are validated. Default = True

Description	Type	Description
ValidateREG	VARIANT_BOOL	Specify whether Windows Registry (.REG) files are validated. Default = True
DoExtensionFiltering	VARIANT_BOOL	Enable extension filtering. Default = False
ExtensionFilteringScope	FileExtensionFilteringScope	Specify whether the file extensions in the FileExtensions property are included or excluded from rules processing. Default = Exclude
FileExtensions	FileExtensionDictionary	A list of extensions used for extension filtering.
ApplicationAccessEnabled	VARIANT_BOOL	Specify whether Application Access Control is enabled. Default = True.
ANACEnabled	VARIANT_BOOL	Specify whether Application Network Access control is enabled. Default = True.
URMEnabled	VARIANT_BOOL	Specify whether User Privileges Management is enabled. Default = True.
IgnoreRestrictionsDuringActiveSetup	VARIANT_BOOL	Ignore restrictions during active setup. Default = False.

Description	Type	Description
ProhibitFilesOnRemovableMedia	VARIANT_BOOL	Prohibit files on removable media. Default = True.

Object: Device

Property	Type	Description
Host	BSTR	The host address.
HostType	DeviceType	Specify whether the address refers to a computer or a connecting device. Default = Computer
NameType	HostNameType	Specify whether the address is a host name or IP address. Default = HostName

Object: DeviceRule

Description	Type	Description
Devices	DeviceDirectory	Collection of devices to which this rule applies.
Name	BSTR	The name of the rule.
SecurityLevel	SecurityLevel	The level of restriction applied to this rule.
AccessibleApplicationGroups	ApplicationGroupReferenceDictionary	Collection of accessible Application Groups.
AccessibleFiles	FileCollection	Collection of allowed files.
AccessibleFolders	FolderCollection	Collection of allowed folders.
AccessibleDrives	DriveCollection	Collection of allowed drives.
AccessibleSignatures	SignatureFileCollection	Collection of allowed signatures.
AccessibleNetworkConnections	NetworkConnectionCollection	Collection of allowed network connections

Description	Type	Description
ProhibitedApplicationGroups	ApplicationGroupReferenceDictionary	Collection of denied Application Groups.
ProhibitedFiles	FileCollection	Collection of denied files.
ProhibitedFolders	FolderCollection	Collection of denied folders
ProhibitedDrives	DriveCollection	Collection of denied drives.
ProhibitedSignatures	SignatureFileCollection	Collection of denied signatures.
ProhibitedNetworkConnections	NetworkConnectionCollection	Collection of denied network connections.

Object: DigitalCertificate

Property	Type	Description
Path	BSTR	Unused for this object.
Description	BSTR	The description of the digital certificate.
EnforceExpiryDate	VARIANT_BOOL	Specify whether the expiry date verification will be applied to this certificate. Default = False
RawCertificateData	BSTR	The base64 encoded digital certificate.
ExpiryDate	BSTR	The certificate expiry date.
ErrorIgnoreFlags	LONG	A bitwise OR operation of the ErrorIgnoreFlags values below. Default = 0

ErrorIgnoreFlags

CERT_CHAIN_POLICY_IGNORE_NOT_TIME_VALID_FLAG 0x00000001

CERT_CHAIN_POLICY_IGNORE_CTL_NOT_TIME_VALID_FLAG 0x00000002

CERT_CHAIN_POLICY_IGNORE_NOT_TIME_NESTED_FLAG 0x00000004

CERT_CHAIN_POLICY_IGNORE_INVALID_BASIC_CONSTRAINTS_FLAG 0x00000008

CERT_CHAIN_POLICY_ALLOW_UNKNOWN_CA_FLAG 0x00000010

CERT_CHAIN_POLICY_IGNORE_WRONG_USAGE_FLAG 0x00000020

CERT_CHAIN_POLICY_IGNORE_INVALID_NAME_FLAG 0x00000040

CERT_CHAIN_POLICY_IGNORE_INVALID_POLICY_FLAG 0x00000080

CERT_CHAIN_POLICY_IGNORE_END_REV_UNKNOWN_FLAG 0x00000100

CERT_CHAIN_POLICY_IGNORE_CTL_SIGNER_REV_UNKNOWN_FLAG 0x00000200

CERT_CHAIN_POLICY_IGNORE_CA_REV_UNKNOWN_FLAG 0x00000400

CERT_CHAIN_POLICY_IGNORE_ROOT_REV_UNKNOWN_FLAG 0x00000800

Object: Drive

Property	Type	Description
Path	BSTR	Full path to drive.
Description	BSTR	The drive description.

Object: File

Property	Type	Description
Path	BSTR	Full path to file.
Description	BSTR	The file description.
Arguments	BSTR	The command line arguments used for spawning a process.
CommandLine	BSTR	The full command line (Path + Arguments) when a file is run.
ApplyAccessTimes	VARIANT_BOOL	Specify whether access times are to be applied. Default = False
AccessTimes	AccessTimes	Collection of access times to be applied.
TrustedOwnershipChecking	VARIANT_BOOL	Specify whether the file is subject to Trusted Ownership checking. Default = True
ApplicationLimit	LONG	The number of concurrent instances of this file that can be executed (0 means unlimited). Default = 0

Object: FileExtension

Property	Type	Description
Name	BSTR	File Extension.

Object: FileMetaData

Description	Type	Description
ProductVersionMaximum	BSTR	The maximum product version number to match.
ProductVersionMaximumEnabled	VARIANT_BOOL	Enables/Disables the use of the ProductVersionMaximum property.
ProductVersionMinimum	BSTR	The minimum product version number to match.
ProductVersionMinimumEnabled	VARIANT_BOOL	Enables/Disables the use of the ProductVersionMinimum property.
FileVersionMaximum	BSTR	The maximum file version number to match.
FileVersionMaximumEnabled	VARIANT_BOOL	Enables/Disables the use of the FileVersionMaximum property.
FileVersionMinimum	BSTR	The minimum file version number to match. Format is <major>.<minor>.<build>.<revision> where each element is a number or the '*' wildcard character to match anything.
FileVersionMinimumEnabled	VARIANT_BOOL	Enables/Disables the use of the FileVersionMinimum property.
VendorName	BSTR	The Vendor Name to match against. Wildcard characters '*' and '?' are supported to match any substring or single character.
VendorNameEnabled	VARIANT_BOOL	Enables/Disables the use of the VendorName property.
ProductName	BSTR	The Product Name to match against. Wildcard characters '*' and '?' are supported to match any substring or single character.
ProductNameEnabled	VARIANT_BOOL	Enables/Disables the use of the ProductName property.
CompanyName	BSTR	The Company Name to match against. Wildcard

Description	Type	Description
		characters '*' and '?' are supported to match any substring or single character.
CompanyNameEnabled	VARIANT_BOOL	Enables/Disables the use of the CompanyName property.
FileDescription	BSTR	The File Description to match against. Wildcard characters '*' and '?' are supported to match any substring or single character.
FileDescriptionEnabled	VARIANT_BOOL	Enables/Disables the use of the FileDescription property.

ObjectFolder

Property	Type	Description
Path	BSTR	Full path to folder.
Description	BSTR	The folder description.
ApplyAccessTimes	VARIANT_BOOL	Specify whether access times are to be applied.
AccessTimes	AccessTimes	Collection of access times to be applied.
TrustedOwnershipChecking	VARIANT_BOOL	Specify whether the folder is subject to Trusted Ownership checking. Default = True
Recursive	VARIANT_BOOL	Whether rules are applied to sub-folders. Default = True

Object: GroupRule

Description	Type	Description
DisplayName .	BSTR	The account name
SID.	BSTR	The account SID
Name	BSTR .	The name of the rule
SecurityLevel	SecurityLevel	The level of restriction applied to this rule.
Groups	ApplicationGroupReferenceDictionary	Collection of allowed

Description	Type	Description
		Application Groups.
AccessibleFiles	FileCollection	Collection of allowed files.
AccessibleFolders	FolderCollection	Collection of allowed folders.
AccessibleDrives	DriveCollection	Collection of allowed drive.
AccessibleSignatures	SignatureFileCollection	Collection of allowed signatures.
AccessibleNetworkConnections	NetworkConnectionCollection	Collection of allowed network connections.
ProhibitedApplicationGroups	ApplicationGroupReferenceDictionary	Collection of denied Application Groups.
ProhibitedFiles	FileCollection	Collection of denied files.
ProhibitedFolders	FolderCollection	Collection of denied folders.
ProhibitedDrives	DriveCollection	Collection of denied drives.
ProhibitedSignatures	SignatureFileCollection	Collection of denied signatures.
ProhibitedNetworkConnections	NetworkConnectionsCollection	Collection of denied network connections.
TrustedVendors	DigitalCertificateCollection	Collection of trusted vendors' digital certificates.
UserRightsRules	URMRules	Configured settings for User Privileges rules.

Object: MessageSettings

Property	Type	Description
DisplayInitialWarningMessage	VARIANT_BOOL	Determines whether the user should be warned that an application is about to be closed due to its allowed time having expired.
CloseApplication	VARIANT_BOOL	Determine whether an application with an expired allowed time should be sent a WM_CLOSE to allow the user chance to save work.
TerminateApplication	VARIANT_BOOL	Determine whether an application with an expired allowed time should be forcefully terminated.
WaitTime	LONG	The delay period between warning the user, sending a WM_CLOSE and terminating the application. This value is in seconds.
AccessDeniedMessageCaption	BSTR	The caption for the denied message box.
AccessDeniedMessageBody	BSTR	The text for the denied message box.
ApplicationLimitsExceededMessageCaption	BSTR	The caption for the message box that is displayed when an application has reached its application limit.
ApplicationLimitsExceededMessageBody	BSTR	The text for the message box that is displayed when an application has reached its application limit.
TimeLimitsWarningMessageCaption	BSTR	The caption for the message box that is displayed when an application has reached the end of its allowed time.
TimeLimitsWarningMessageBody	BSTR	The text for the message box that is displayed when an application has reached the end of its allowed time.
TimeLimitsDeniedMessageCaption	BSTR	The caption for the message box that is displayed when an application is denied due to a time restriction.

Property	Type	Description
TimeLimitsDeniedMessageBody	BSTR	The text for the message box that is displayed when an application is denied due to a time restriction.
SelfAuthorizationMessageCaption	BSTR	The caption for the message box that is displayed when user authorization is required to run a file.
SelfAuthorizationMessageBody	BSTR	The text for the message box that is displayed when user authorization is required to run a file.
SelfAuthorizationResponseCaption	BSTR	The text for the message box that is displayed when the user has previously self-authorized a file to run.
SelfAuthorizationResponseBody	BSTR	The caption for the message box that is displayed when the user has previously self-authorized a file to run.

Object: NetworkConnection

Property	Type	Description
Path	BSTR	Full path to network resource.
Description	BSTR	The description of the network resource.
Address	BSTR	The address of the network resource, for example, www.bbc.co.uk.
Resource	BSTR	The resource path, for example \weather.
Port	BSTR	The port to which this network connection applies, if appropriate.
UseWildcards	VARIANT_BOOL	Specify whether any part of the whole network location contains wildcards.
AddressType	NetworkConnectionType	The connection type. Default = False
Recursive	VARIANT_BOOL	Specify whether child resources are included as part of this connection.

Object: OnDemandConfigChangeSettings

Property	Type	Description
OnDemandEnabled	VARIANT_BOOL	Global On/Off for Policy Change Request. Default = False
EmailRequestsEnabled	VARIANT_BOOL	Enables the Email Request functionality for Policy Change Requests. Default = True.
MailToAddress	BSTR	BSTR Specifies the Recipient Email Address
EmergencyRequestsEnabled	VARIANT_BOOL	Enables the Immediate Change Request functionality. Default = True.
HelpDeskPhoneNumber	BSTR	Specifies the phone number for the Help Desk.
SharedKey	BSTR	Specifies the salt for use in encryption algorithms. Must use ASCII characters and match the key used by the Help Desk. This is to be used in conjunction with the ConfigurationHelper object. For further information, see Policy Change Request .

Property	Type	Description
RequestMethods	OnDemandConfigChangeUserInteractionSet up	Configures the request methods.

Object: OnDemandConfigChangeUserInteractionSetup

Property	Type	Description
AllowLinkFromAMDenied	VARIANT_ BOOL	Enable link through from AMDenied Message. Default = True.
AMDeniedLinkText	BSTR	Specify the text displayed in the AMDenied. Message dialog link.
ShowShellMenu	VARIANT_ BOOL	Enables the right-click context option menu. Default = True.
ShellMenuText	BSTR	Specify the text displayed in the right-click context menu.
ShowDesktopIcon	VARIANT_ BOOL	Enables the Policy Change Request desktop icon. Default = True.
DesktopIconText	BSTR	Specify the text displayed on the Policy Change Request desktop icon.

Object: ProcessRule

Property	Type	Description
SecurityLevel	SecurityLevel	The level of restriction applied to this rule.
AccessibleApplicationGroups	ApplicationGroupReferenceDictionary	Collection of allowed Application Groups.
AccessibleFiles	FileCollection	Collection of allowed files.
AccessibleFolders	FolderCollection	Collection of allowed folders.
AccessibleDrives	DriveCollection	Collection of allowed drive.
AccessibleSignatures	SignatureFileCollection	Collection of allowed

Property	Type	Description
		signatures.
AccessibleNetworkConnections	NetworkConnectionCollection	Collection of allowed network connections.
ProhibitedApplicationGroups	ApplicationGroupReferenceDictionary	Collection of denied Application Groups.
ProhibitedFiles	FileCollection	Collection of denied files.
ProhibitedFolders	FolderCollection	Collection of denied folders.
ProhibitedDrives	DriveCollection	Collection of denied drives.
ProhibitedSignatures	SignatureFileCollection	Collection of denied signatures.
ProhibitedNetworkConnections	NetworkConnectionsCollection	Collection of denied network connections.
TrustedVendors	DigitalCertificateCollection	Collection of trusted vendors' digital certificates.
UserRightsRules	URMRules	Configured settings for User Privileges rules.
FileProcessItems	FileCollection	Collection of processes to which this rule applies.
SignatureProcessItems	SignatureProcessItems	Collection of processes to which this rule applies, defined by signature.

Object: ScriptedRule

Property	Type	Description
EntryFunction	BSTR	The function that will be executed when the

Property	Type	Description
		script is launched.
Script	BSTR	The body of the script.
Context	ExecutionContext	The context in which the script executed. Default = PerSessionAsUser.
WaitForLogin	VARIANT_BOOL	Specify whether the execution of the script will be delayed until the login process is complete. Default = False
Timeout	LONG	The timeout period a script is given before being terminated.
Name	BSTR	The name of the rule.
SecurityLevel	SecurityLevel	The level of restriction applied to this rule.
AccessibleApplicationGroups	ApplicationGroupReferenceDictionary	Collection of allowed Application Groups.
AccessibleFiles	FileCollection	Collection of allowed files.
AccessibleFolders	FolderCollection	Collection of allowed folders.
AccessibleDrives	DriveCollection	Collection of allowed drive.
AccessibleSignatures	SignatureFileCollection	Collection of allowed signatures.
AccessibleNetworkConnections	NetworkConnectionCollection	Collection of allowed network connections.
ProhibitedApplicationGroups	ApplicationGroupReferenceDictionary	Collection of denied Application Groups.

Property	Type	Description
ProhibitedFiles	FileCollection	Collection of denied files.
ProhibitedFolders	FolderCollection	Collection of denied folders.
ProhibitedDrives	DriveCollection	Collection of denied drives.
ProhibitedSignatures	SignatureFileCollection	Collection of denied signatures.
ProhibitedNetworkConnections	NetworkConnectionsCollection	Collection of denied network connections.
TrustedVendors	DigitalCertificateCollection	Collection of trusted vendors' digital certificates.
UserRightsRules	URMRules	Configured settings for User Privileges rules.
FileProcessItems	FileCollection	Collection of processes to which this rule applies.
SignatureProcessItems	SignatureProcessItems	Collection of processes to which this rule applies, defined by signature.

Object: SignatureFile

Property	Type	Description
Path	BSTR	Full path to the file.
Description	BSTR	The file description.
Arguments	BSTR	The command line arguments used for spawning a process.
SHA1 Hash	BSTR	The SHA1 hash of the file.
CommandLine	BSTR	The full command line (Sha1Hash + Arguments) when a file is run.

Property	Type	Description
Version	BSTR	The file version information.
ApplyAccessTimes	VARIANT_BOOL	Specify whether access time are to be applied. Default = False
AccessTimes	AccessTimes	Collection of access times to be applied.

Object: TimeRange

Property	Type	Description
StartHour	LONG	The hour at which the time range starts.
EndHour	LONG	he hour at which the time range ends.

Object: TrustedOwner

Property	Type	Description
DisplayName	BSTR	The account name.
SID	BSTR	The account SID.
Description	BSTR	The account description.

Object: URMGroupBehaviour

DisplayName B	STR	The name of the group.
SID	BSTR	The group's SID.
Action	URMGroupAction	The action to perform with this group. Default = Add

Object: URMPolicy

Property	Type	Description
Name	BSTR	Name of the policy.
Description	BSTR	A description for the policy.
GroupMembershipActions	URMGroupBehaviourDictionary	A collection of configured UPM (User Privilege Management) Group Behavior actions.

Property	Type	Description
PrivilegeActions	URMPrivilegeDictionary	A collection of configured UPM Privilege actions.

Object: URMPrivilege

Property	Type	Description
Name	BSTR	Textual description of the privilege.
Privilege	URMPrivilegeConstant	The privilege being set. Default = SeAssignPrimaryTokenPrivilege
Action	URMPrivilegeAction	The action to perform on the privilege Default = NoChange.

Object: URMRuleItem

Property	Type	Description
KeyPath	BSTR	The keypath used in collections of URMRuleItems.
Application	RuleItem	The application for which to apply the User Rights setting. Can be of type File, Folder, Signature File or Application Group.
ApplyToChildren	VARIANT_BOOL	Setting to specify whether the user rights setting should be applied to any child processes. Default = False.

Object: URMRuleItemPolicy

Property	Type	Description
KeyPath	BSTR	The keypath used in collections of URMRuleItems.
Application	RuleItem	The application to which to apply the User Rights policy. Can be of type File, Folder, Signature File or Application Group.
ApplyToChildren	VARIANT_BOOL	Setting to specify whether the user rights policy should be applied to any child processes. Default = False.
Policy	URMPolicyReference	The URM Policy to apply to the application.

Object: URMRules

Property	Type	Description
URMFiles	URMRuleItemPolicyDictionary	Collection of files and User Privileges Management (UPM) policies to apply to them.
URMSignatures	URMRuleItemPolicyDictionary	Collection of signature files and UPM policies to apply to them.
URMFolders	URMRuleItemPolicyDictionary	Collection of folders and UPM policies to apply to them.
URMApplicationGroups	URMRuleItemPolicyDictionary	Collection of Application Groups and UPM policies to apply to them.

Object: UserRule

Property	Type	Description
DisplayName	BSTR	The account name.
SID	BSTR	The account SID.
Name	BSTR	The name of the rule.
SecurityLevel	SecurityLevel	The level of restriction applied to this rule.
AccessibleApplicationGroups	ApplicationGroupReferenceDictionary	Collection of allowed Application Groups.
AccessibleFiles	FileCollection	Collection of allowed files.
AccessibleFolders	FolderCollection	Collection of allowed folders.
AccessibleDrives	DriveCollection	Collection of allowed drive.
AccessibleSignatures	SignatureFileCollection	Collection of allowed signatures.
AccessibleNetworkConnections	NetworkConnectionCollection	Collection of allowed network connections.

Property	Type	Description
ProhibitedApplicationGroups	ApplicationGroupReferenceDictionary	Collection of denied Application Groups.
ProhibitedFiles	FileCollection	Collection of denied files.
ProhibitedFolders	FolderCollection	Collection of denied folders.
ProhibitedDrives	DriveCollection	Collection of denied drives.
ProhibitedSignatures	SignatureFileCollection	Collection of denied signatures.
ProhibitedNetworkConnections	NetworkConnectionsCollection	Collection of denied network connections.
TrustedVendors	DigitalCertificateCollection	Collection of trusted vendors' digital certificates.
UserRightsRules	URMRules	Configured settings for User Privileges rules.

Enumerations

Name: Device Type

Computer = 0

ConnectingDevice = 1

Name: ExecutionContext

PerSessionAsUser = 0

PerSessionAsSystem = 1

PerComputerAsSystem = 2

Name: FileExtensionFilteringScope

Exclude = 0

Include = 1

Name: HostNameType

HostName = 0

IPAddress = 1

Name: NetworkConnectionType

HostAddress = 0

IPAddress = 1

UNCPath = 2

Name: ScriptingLanguage

VBScript = 0

PowerShell = 1

Name: SecurityLevel

Restricted = 0

SelfAuthorizing = 1

Unrestricted = 2

AuditOnly = 3

Configuration Helper Object

The Configuration Helper object provides useful functionality that is not provided by the configuration model, such as the ability to load and save configurations.

The methods listed below provide error reporting as a HRESULT which can be tested for in VBScript using the Err object. Success is reported as S_OK which is 0.

In case of error, most of the time the Configuration Helper Object returns the error code 2147500037 which is 0x80004005 in hex and defined as E_FAIL in COM. The other most common error is 2147942405 which is 0x80070005 in hex and defined as

Configuration Helper Object Methods

LoadLiveConfiguration (method)

Returns:

BSTR - The xml representation of the live configuration.

HRESULT - Returns S_OK if successful.

SaveLiveConfiguration (method)**Returns:**

HRESULT - Returns S_OK if successful.

Parameters:

BSTR - The xml representation of the configuration loaded from disk.

LoadLocalConfiguration (method)**Returns:**

BSTR - The xml representation of the configuration loaded from disk.

HRESULT - Returns S_OK if successful.

Parameters:

BSTR - The full file path of the configuration to load.

SaveLocalConfiguration (method)**Parameters:**

BSTR - The full file path of the configuration to load.

BSTR - The xml representation of the configuration to save.

ReadNumCertificatesFromFile (method)**Returns:**

LONG - The number of certificates used to sign the specified executable file.

Parameters:

BSTR - The full file path of the executable file used in determining the certificate count.

ReadCertificateFromFile (method)**Returns:**

BSTR - The raw certificate data.

Parameters:

BSTR - The full file path of the executable file from which the certificate will be read.

LONG - The index of the certificate to read.

ReadSha1HashFromFile (method)

Returns:

BSTR - The hash value.

HRESULT - Returns S_OK if successful.

Parameters:

BSTR - The full file path of the file for which the hash will be generated.

DefaultConfiguration (property)

This BSTR property contains the xml representation of the default configuration.

The DefaultConfiguration() method only returns a configuration in the English language. This means that some group names and other text in the configuration may not be in the native language of the operating system, which can result in the configuration not being applied correctly. For non-English operating systems it is necessary to export the default configuration from the product console on a native operating system. This can be stored as a file on the network or distributed to the machine where the configuration scripting will be performed. Once this is done, use the LoadLocalConfiguration() method in place of the DefaultConfiguration(). This will produce the same configuration but in the correct native language.

LoadLocalConfigurationWithAuditing (method)

Returns:

BSTR - The xml representation of the live configuration

BSTR - The xml representation of the live Auditing configuration

HRESULT - Return S_OK if successful

Parameters:

BSTR - The full file path of the configuration to load

SaveLocalConfigurationWithAuditing (method)

Parameters:

BSTR - The full file path of the configuration to save

BSTR - The xml representation of the configuration to save

BSTR - The xml representation of the auditing configuration to save

SaveLocalConfigurationWithAuditingFile (method)**Parameters:**

BSTR - The full file path of the configuration to save

BSTR - The xml representation of the configuration to save

BSTR - The full file path of the Auditing.xml to save

LoadLiveConfigurationWithAuditing (method)**Returns:**

BSTR - The xml representation of the live configuration

BSTR - The xml representation of the live Auditing configuration

HRESULT - Return S_OK if successful

SaveLiveConfigurationWithAuditing (method)**Parameters:**

BSTR - The xml representation of the configuration to save

BSTR - The xml representation of the auditing configuration to save

SaveLiveConfigurationWithAuditingFile (method)**Parameters:**

BSTR - The xml representation of the configuration to save

BSTR - The full file path of the Auditing.xml to save

EncryptSharedKey (method)**Parameters:**

BSTR - The shared key used in Policy Change Requests

Returns:

BSTR - Encrypted version of the shared key

HRESULT - Return S_OK if successful

LoadLocalConfigurationHandle

Parameters:

BSTR - The full file path of the configuration to load

Returns:

VARIANT - Opened file handle

BSTR - The xml representation of the configuration

HRESULT - Return S_OK if successful

LoadLiveConfigurationHandle

Returns:

VARIANT - Opened file handle

BSTR - The xml representation of the live configuration

HRESULT - Return S_OK if successful

LoadLocalConfigurationHandleWithAuditing

Parameters:

BSTR - The full file path of the configuration to load

Returns:

BSTR - The xml representation of the configuration

BSTR - The xml representation of the auditing configuration

VARIANT - Opened file handle

HRESULT - Return S_OK if successful

LoadLiveConfigurationHandleWithAuditing

Returns:

BSTR - The xml representation of the configuration

BSTR - The xml representation of the auditing configuration

VARIANT - Opened file handle

HRESULT - Return S_OK if successful

SaveLocalConfigurationHandle

Parameters:

BSTR - The full file path of the configuration to save

BSTR - The xml representation of the configuration

VARIANT - Opened file handle

SaveLiveConfigurationHandle

Parameters:

BSTR - The xml representation of the configuration

VARIANT - Opened file handle

SaveLocalConfigurationHandleWithAuditing

Parameters:

BSTR - The full file path of the configuration to save

BSTR - The xml representation of the configuration

BSTR - The xml representation of the auditing configuration

VARIANT - Opened file handle

SaveLiveConfigurationHandleWithAuditing

Parameters:

BSTR - The xml representation of the configuration

BSTR - The xml representation of the auditing configuration

VARIANT - Opened file handle

Import and Export Scripted Rules

The Scripted Rule import and export feature copies PowerShell and VBScripts from one Application Control console to another and enables you to import a script that has been written in another editor.

Export a Scripted Rule

1. Navigate to the Scripted Rule node and select the rule to be exported.
2. In the Current Script section of the work area, select **Click here to edit script**.
The Configure this Scripted Rule dialog displays
3. Ensure the script displayed is correct.
4. Click **Export**.
5. Navigate to where you want the script to be exported and click **Save**.
6. Click **OK**.

Import a Scripted Rule

When you import a script, any existing script displayed in the Configure this Scripted Rule dialog is overwritten.

1. Navigate to the Scripted Rule node and select where the rule is to be imported.
2. In the Current Script section of the work area, select **Click here to edit script**.
The Configure this Scripted Rule dialog displays
3. Click **Import**.
4. Navigate to where you saved the script and click **Open**.
5. Click **OK**.

Appendix

Citrix XenApp

To set up Citrix XenApp streaming applications to work with certain elements of Application Control, you need to specify certain exclusions, as follows:

1. Navigate to Citrix Streaming Profiler for Windows.
2. Open the Application Profile.
3. Highlight the relevant Target and select the Edit menu.
4. Select **Target Properties**.

The Target Properties screen displays.

5. Select **Rules**.

The Rules work area displays.

6. Click **Add** in the Rules work area.

The New Rule Select Action and Objects dialog displays.

7. In the Action section leave the default setting as **Ignore**.

8. In the Object section select Named Objects and click **Next**.

The New Rule Select Objects dialog displays.

9. Select **Some Named Objects** and click **Add**.

The Choose Named Object dialog displays.

10. Add `\\??\pipe\Appsense*` and click **OK**.

This displays in Named Objects on the New Rule Select Objects dialog.

11. Click **Next** to display the New Rule Name Rule dialog.

12. Enter a name for the rule or accept the default and click **Finish**.

13. Click **OK**.

The Target Properties screen displays and the Ignore all named objects rule is listed in the work area.

14. Save the Profile.

Repeat for each Application Profile as required.

Web Services Configuration

Prerequisites

The system requirements for Application Manager Web Services are:

- Microsoft .NET Framework 4.0 Full (x86 and x64)
- Microsoft Visual C++ 2015 x86 Redistributable package. This is required for both x64 and x86 versions of Application Control.



For further information on required utilities and components, see the [User Workspace Manager Install and Configure Guide](#).

Web Services Port Configuration

The Application Manager Web Service provides two communication routes:

- With machines hosting the Application Control Agent to allow reporting of data.
- With the Application Control Console to allow querying of collected data.

Communication with the Application Manager Web Service is via HTTP or optionally Secure HTTP (HTTPS), defaulting to the standard TCP ports 80 for HTTP and 443 for HTTPS. It is recommended that you use the default values, as these ports are already well known by firewall products and should provide the most trouble-free installation.

However, should you find you have port conflict with other software, follow the steps to configure the Application Manager Web Service to use ports that are free.

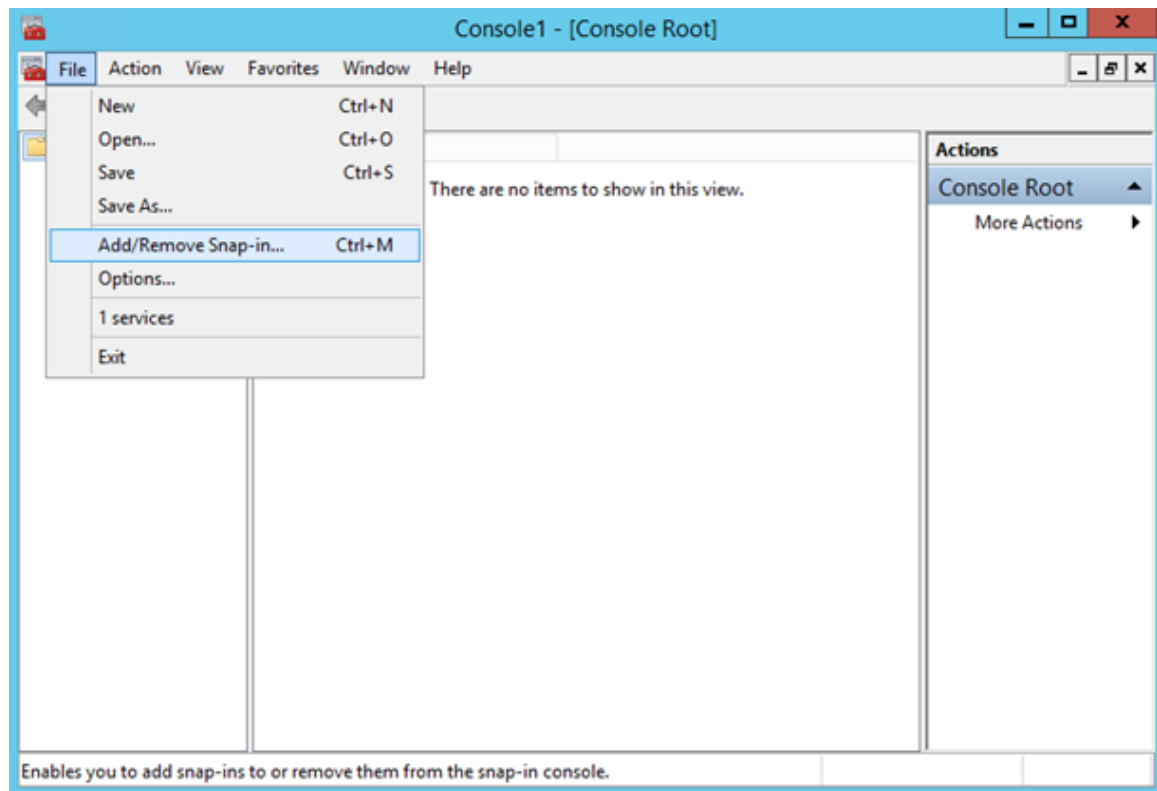
Configure Application Manager Web Services to use SSL

This process describes how to configure the Application Manager Web Services to use secure sockets for communication.

1. Click **Start > Run** and enter **MMC**.

The Microsoft Management Console displays.

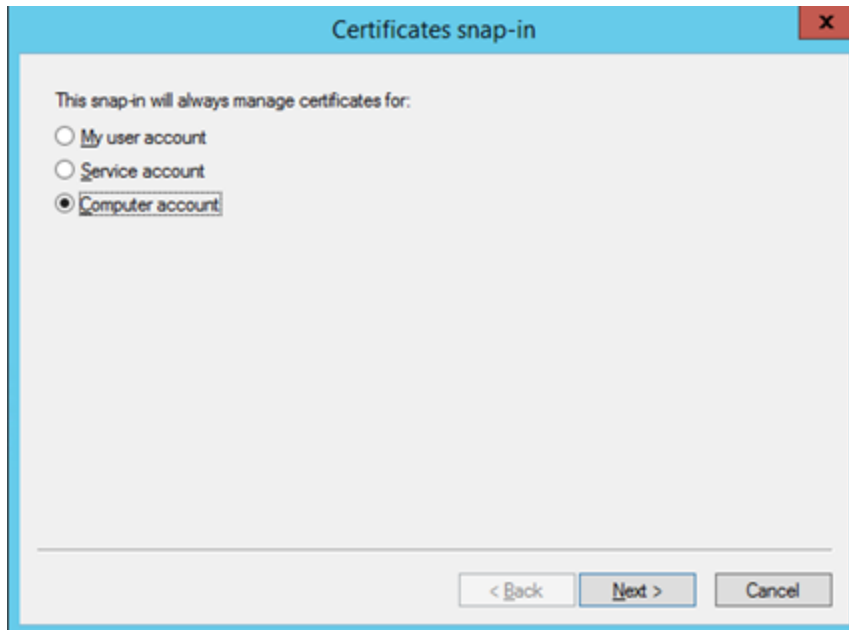
2. Click **File** > **Add/Remove Snap-in...**



The Add or Remove Snap-ins dialog displays.

3. Select **Certificates** and click **Add**.

- From the Certificates snap-in dialog, select **Computer account** and click **Next**.



- Click **Finish** and then **OK**.

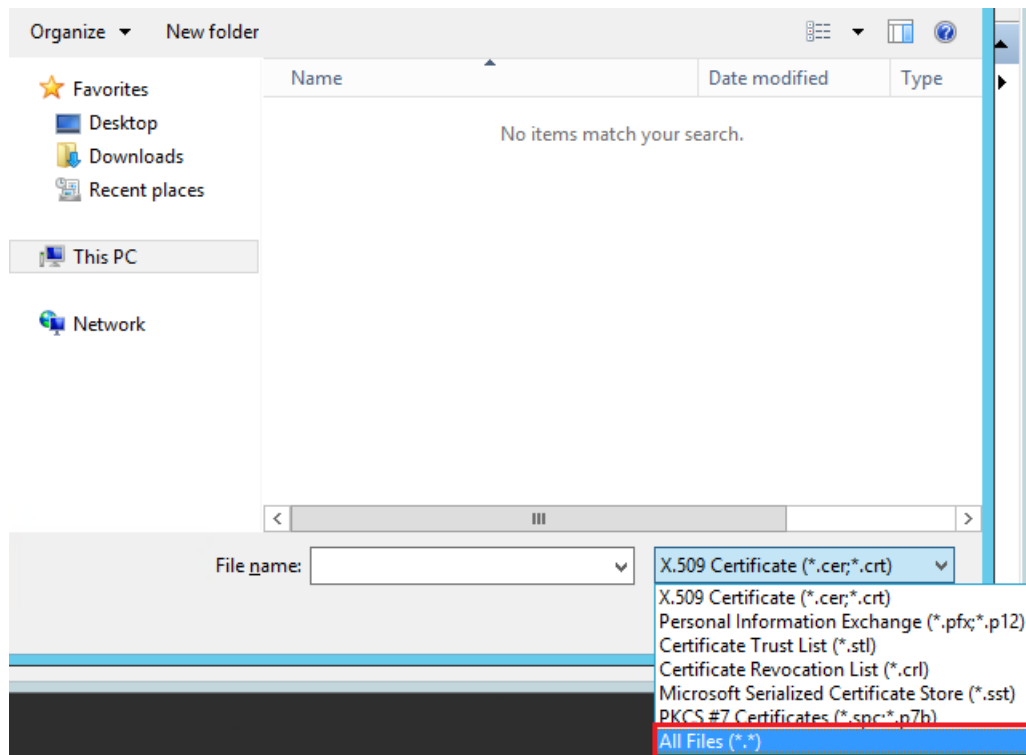
The snap-in is added to the MMC.

- From the navigation tree, select **Certificates (Local Computer) > Personal**.
- Right-click **Personal** and select **All Tasks > Import...**

The Certificate Import Wizard displays.

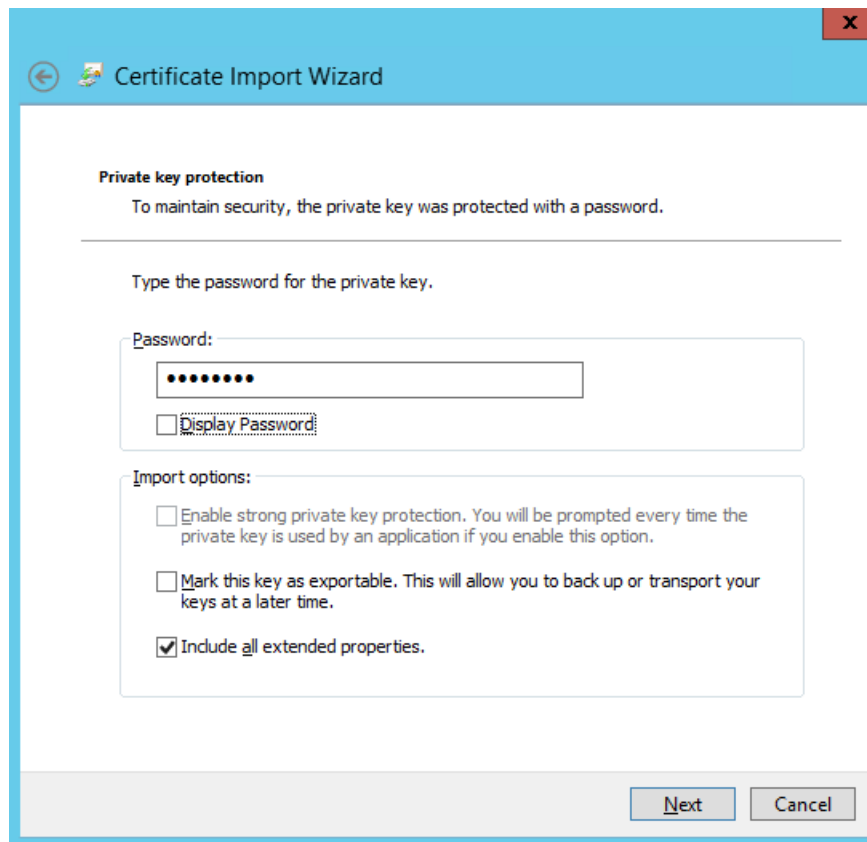
- Click **Next**.

- Click **Browse** and change select **All Files** in the Open dialog.



- Navigate to, and select, the required **PFX** file and click **Open**.
- Click **Next**.

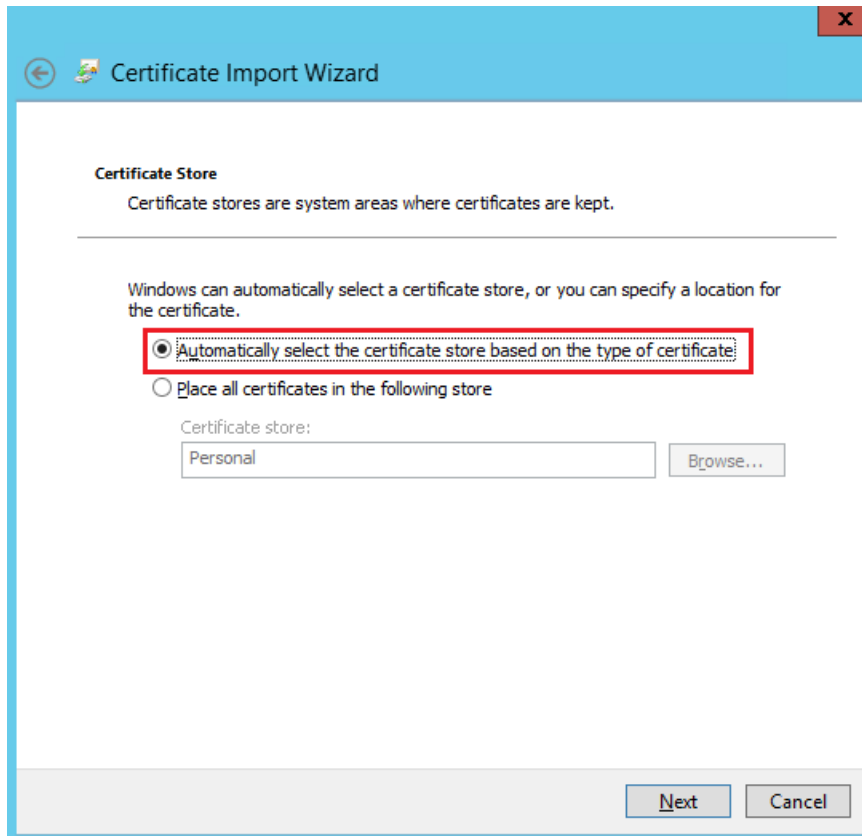
12. Enter the password for the private key and click **Next**.



The image shows a Windows-style dialog box titled "Certificate Import Wizard". It has a blue header bar with a back arrow icon on the left and a close button (X) on the right. The main content area is white and contains the following elements:

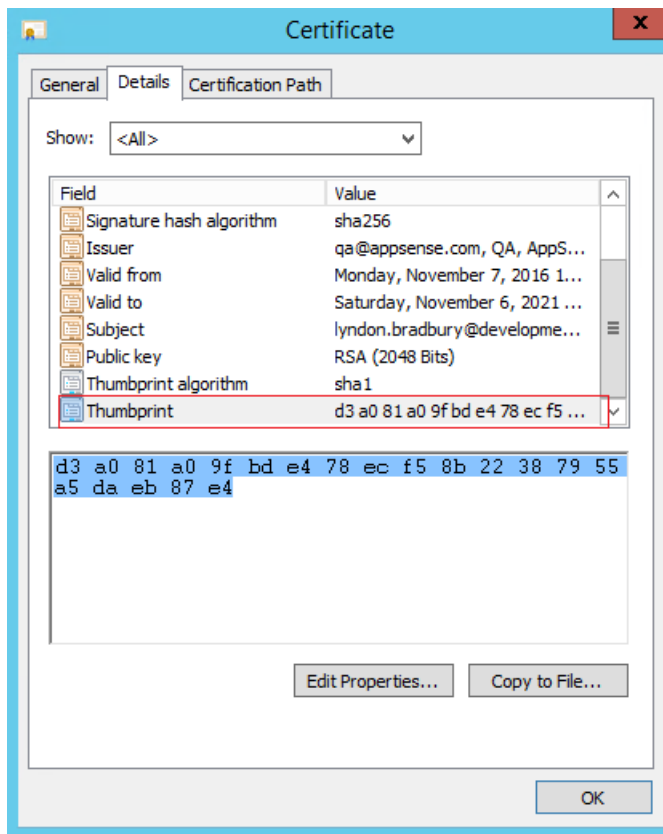
- Private key protection**
To maintain security, the private key was protected with a password.
- A horizontal line separator.
- The text: "Type the password for the private key."
- A "Password:" label followed by a text input field containing eight black dots.
- A checkbox labeled "Display Password" which is currently unchecked.
- An "Import options:" label followed by a group box containing three options:
 - ☐ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.
 - ☐ Mark this key as exportable. This will allow you to back up or transport your keys at a later time.
 - ☒ Include all extended properties.
- At the bottom right, there are two buttons: "Next" and "Cancel".

13. Select **Automatically select the certificate store based on the type of certificate** option and click **Next**.

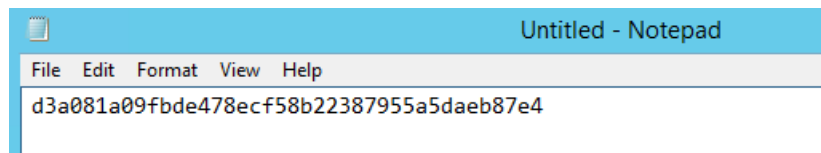


14. Click **Next** then **OK** to complete the import and close the wizard.
15. Refresh the MMC.
The certificate displays in the Personal > Certificates store.
16. Right-click on the machine certificate and select **Open**.
17. Select the **Details** tab.

18. Select **Thumbprint** to display the value in the text box below.



19. Copy the value and paste it into a text editor, such as Notepad.
20. Remove any spaces from the value. This will be used for the certhash value in the commands entered in step 26.



21. Click **OK** to close the certificate.
22. Close MMC without saving.
23. Stop Application Manager Web Services.
24. From an elevated Notepad, open the following file:
- ```
%ProgramFiles
(x86)%AppSense\ApplicationManager\AnalysisService\AnalysisServiceCore.dll.config
```
25. In the file, change `http://localhost:80/ondemand` to `https://localhost:443/ondemand` and save.

26. From an elevated CMD on the server, run the following commands, replacing the certhash values with your thumbprint value from step 20:
  - `netsh http add sslcert hostnameport=localhost:443  
certhash=d3a081a09fbde478ecf58b22387955a5daeb87e4 appid=  
{00000000-0000-0000-0000-000000000000} certstorename=my`
  - `netsh http add sslcert hostnameport=lb-svr2012-r2-5:443  
certhash=d3a081a09fbde478ecf58b22387955a5daeb87e4 appid=  
{00000000-0000-0000-0000-000000000000} certstorename=my`
27. Start Application Manager Web Services.
28. From a browser, test the connection to the web service using https: `https:// lb-svr2012-r2-5/ondemand`
29. Authenticate with a valid user.

## Configuring TCP port numbers used for Communication

For the two communication routes you can independently configure the ports used for HTTP and HTTPS, meaning up to four different port numbers could be configured.

However, for simplicity it is recommended that if you are changing port configuration that you make the same changes to both communication routes. If you are choosing to change the port used for HTTP, then make the change for HTTP on both routes and similarly if changing the Secure HTTP port.



Any firewalls on the machines participating in communication must allow connections over the configured ports.

---

### Quick Setup

Use the following procedure to configure your ports.

1. Open an administrator level Command Prompt and type: `netsh http show urlacl`

A list of the reserved URLs displays.

2. Verify that the following entries exist:

Reserved URL: `https://+:443/AmAnalysisService/`

User: NT AUTHORITY\LOCAL SERVICE

Listen: Yes

Delegate: No

SDDL: D:(A;;GX;;;LS)

Reserved URL : `https://+:443/AmAnalysisQueryDataService/`

User: NT AUTHORITY\LOCAL SERVICE

Listen: Yes

Delegate: No

SDDL: D:(A;;GX;;;LS)

Reserved URL : `https://+:443/OnDemand/`

User: NT AUTHORITY\LOCAL SERVICE

Listen: Yes

Delegate: No

SDDL: D:(A;;GX;;;LS)

4. Using a text editor, open the AMAnalysisServiceCore.dll.config located in:

```
C:\Program Files
(x86)\AppSense\ApplicationManager\AnalysisService\AMAnalysisServiceCore.dll.config
```

5. In the text document search for the following statement:

```
<add key="ON_DEMAND_SERVICE_URI"
value="https://localhost:80/OnDemand/" />
```

6. Replace the statement with following:

```
<add key="ON_DEMAND_SERVICE_URI"
value="https://localhost:443/OnDemand/" />
```

7. Save and close the text document.
8. Using Internet Explorer, navigate to <https://localhost:443/OnDemand>.



The text editor must be elevated to save the document.

---

## Configuring the Windows HTTP Subsystem

The Application Manager Web Service uses the Port Sharing feature of Windows HTTP (HTTP.SYS), allowing it to co-exist with other applications making use of the same mechanism. At a basic level this is achieved by an application registering a portion of a URL, including a port number, for which it will be responsible for servicing requests.

The Application Manager Web Service runs under the account of Local Service, so you need to grant access to the port for that account. You can do this using the httpcfg.exe or netsh.exe tool, depending on the OS Version on which you have installed the Application Manager Web Services:

### Running Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 7, Windows 8, and Windows 8.1

Use the Netsh.exe tool to configure and display the status of various network communications server roles and components.

For further information, see [http://technet.microsoft.com/en-us/library/cc754753\(Ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc754753(Ws.10).aspx).

Do the following:

Open an administrator level Command Prompt and type: `netsh http add urlacl sddl=D:  
(A;;GX;;;LS) url=<URL>`

<URL> is the URL containing the port you wish to grant access to.

## Configuring the Application Manager Web Services for Agent Configuration

Use the netsh.exe or httpcfg.exe tool with the following URLs and commands to configure each of the ports. The examples shown here use the netsh.exe tool but the same command parameters are used with httpcfg.exe:

### HTTP Port

Enter `http://+:<HTTP_PORT>/AmAnalysisService/`

The <HTTP\_PORT> in this example, is to represent the port number you have chosen for standard HTTP communication.

For example, to use port 81 enter the command:

```
netsh http add urlacl sddl=D:(A;;GX;;;LS) url=
http://+:81/AmAnalysisService/
```

### Secure HTTP (HTTPS) Port

Enter `https://+:<HTTPS_PORT>/AmAnalysisService/`

The <HTTPS\_PORT> in this example, is to represent the port number you have chosen for Secure HTTP communication.

For example, to use port 444 enter the command:

```
netsh http add urlacl sddl=D:(A;;GX;;;LS) url=
https://+:444/AmAnalysisService/
```

## Configuring the Application Manager Web Services for Console Communication

Use the netsh.exe or httpcfg.exe tool with the following URLs and commands for each of the ports. The examples shown here use the netsh.exe tool but the same command parameters are used with httpcfg.exe:

### HTTP Port

Enter `http://+:<HTTP_PORT>/ AmAnalysisQueryDataService/`

The <HTTP\_PORT> in this example, is to represent the port number you have chosen for standard HTTP communication.

For example, to use Port 81 enter the command:

```
netsh http add urlacl sddl=D:(A;;GX;;;LS) url=
http://+:81/AmAnalysisQueryDataService/
```



## Secure HTTP (HTTPS) Port

Enter `https://+:<HTTPS_PORT>/ AmAnalysisQueryDataService/`

The `<HTTPS_PORT>` in this example, is to represent the port number you have chosen for Secure HTTP communication

For example, to use Port 444 enter the command:

```
netsh http add urlacl sddl=D:(A;;GX;;;LS) url=
https://+:444/AmAnalysisQueryDataService/
```

## Editing the Application Manager Web Services Configuration

Editing the Application Manager Web Service configuration requires manual editing of its XML configuration file - `AMAnalysisServiceCore.dll.config` - located in the directory where the Application Manager Web Services are installed.

Default Installation Directories:

- 32Bit OS - `C:\Program Files\AppSense\Application Manager\AM Web Services`
- 64Bit OS - `C:\Program Files (x86)\AppSense\Application Manager\AM Web Services`



**Caution:** Before starting it is highly recommended that you take a backup copy of this file because an incorrect configuration can prevent the Application Manager Web Services from starting.

---



In the process below, `<HTTP_PORT>` is the port number you have chosen for standard HTTP communication.

---

1. Using the Windows Services Administrative tool, stop the Application Manager Web Service.
2. Open the `AMAnalysisServiceCore.dll.config` file in a text editor such as Notepad and navigate to the `<services>` XML tag.
3. If required, do the following:
  - To reconfigure the Agent to Service communication, go to Step 4.
  - To reconfigure the Console to Service communication, go to step 9.

4. To change the HTTP Port, navigate to the following the XML tag:

```
<service name="AmAnalysisServiceCore.AmAnalysisWebService">
```

5. Select the `webHttpNonSecureBinding` attribute and replace the address value using the following format:

```
address="http://localhost:<HTTP_PORT>/AmAnalysisService"
```

6. To change the Secure HTTP Port, navigate to the following XML tag:

```
<service name="AmAnalysisServiceCore.AmAnalysisWebService">
```

7. Select the `webHttpSecureBinding` attribute and replace the address value using the following format:

```
address="http://localhost:<HTTPS_PORT>/AmAnalysisService
```

8. To change the HTTP Port, navigate to the following the XML tag:

```
<service name="AmAnalysisServiceCore.AmQueryDataWebService">
```

9. Select the `wsHttpNonSecureBinding` attribute and replace the address value using the following format:

```
address="http://localhost:<HTTP_PORT>/AmAnalysisQueryDataService
```

10. To change the Secure HTTP Port, navigate to the following XML tag:

```
<service name="AmAnalysisServiceCore.AmQueryDataWebService">
```

11. Select the `wsHttpSecureBinding` attribute and replace the address value using the following format:

```
address="http://localhost:<HTTP_PORT>/AmAnalysisQueryDataService
```

12. Save and close the `AMAnalysisServiceCore.dll.config` file.



The text editor must be elevated to save the document.

---

13. Using a text editor, open the `AMAnalysisServiceCore.dll.config` located in:

```
C:\Program Files
(x86)\AppSense\ApplicationManager\AnalysisService\AMAnalysisServiceCore.dll.config
```

14. In the text document search for the following statement:

```
<add key="ON_DEMAND_SERVICE_URI"
value="http://localhost:80/OnDemand/" />
```

15. Replace the statement with the following:

```
<add key="ON_DEMAND_SERVICE_URI"
value="http://localhost:443/OnDemand/" />
```

16. Save and close the text document.

17. Using the Windows Services Administrative tool, start the Application Manager Web Service.

If there are any problems starting the Service, refer to the Windows Event Log under AppSense for error data.

## Wildcards and Regular Expressions

Application Control uses regular expressions when you select the Use regular expressions option when adding filenames, folder paths, command line arguments and metadata strings. Regular expressions are not supported for Groups and User Name Rules.



When using regular expressions, you need to check the expression fully before committing it. If the criteria are incorrectly entered, both complete and partial matches are returned.

Metacharacter	Matches
<code>^[a-f]+</code>	"alice" matches because her name starts with a letter between a and f "john" does not match because his name starts with a letter greater than f "Alice" does not match because her name does not start with a lowercase letter
<code>^[a-zA-F]+</code>	"Alice" matches because with this expression uppercase letters are allowed
<code>[a-zA-Z]+\d\d\d\$</code>	"UserWithThreeNumbers123" matches because the user name is made up of letters followed by three numbers. "UserWithFourNumbers1234" does not match because the user name has four numbers in it
<code>(notepad)   (winword)   (calc) .exe</code>	notepad.exe matches because it is in the list wordpad.exe does not match because it is not in the list

The information below shows examples of how regular expression and wildcards can be used in Application Control.

Example	Description
<code>^chkdsk((?!/[f x]).)*\$</code>	Used to allow users to run the check disk utility and view the result, but prevent fixing any errors on the disk using a /f (fix) parameter.

## Distributed File Systems

A distributed file system or network file system allows access to files from multiple hosts sharing via a computer network. This makes it possible for multiple users on multiple machines to share files and storage resource. Using DFS, System administrators can make it easy for users to access and manage files that are physically distributed across a network.

There are two ways of implementing DFS on a server:

- Standalone DFS Namespaces
- Domain-Based DFS Namespaces

For examples of that can be part of both a domain and standalone scenario, see [Choosing the DFS Namespace Type](#).

For Application Network Access Control (ANAC) rules using a network share and files or folders that refer to items on a DFS share, you must specify the target server, rather than the namespace server in the UNC path. Application Control Agent substitutes the namespace server path with the target server path, so the namespace server path never gets passed through the rules engine.

## App-V5.0 Support

App-V 5.0 allows applications to be streamed in real-time to any client from a virtual application server. With a streaming-based implementation, the App-V client needs to be installed on the client machines. Application packages are presented on the App-V server and then streamed to the endpoint cache. At the first application package launch request, the package is streamed to the endpoint. For any subsequent application launch requests made on that specific endpoint the application package will be run from the local App-V client cache.

By default, App-V 5.0 applications are inherently trusted, which means they will not fail the Trusted Ownership check. The Application Control Agent caches any published App-V 5.0 applications at session startup. If any application is published mid-session it will not pass Application Managers Trusted Ownership check until that user logs off and logs on again.

## Components

The following table lists the Windows Control Panel components that can be elevated using the User Privilege Management Components feature.

Component Name	Type	Operating System
Add and Remove Programs/Programs and Features <ul style="list-style-type: none"> <li>Uninstall an Update</li> <li>Uninstall or Change Program</li> </ul>	Control Panel	All O/S
Add Hardware	Control Panel	All O/S
Automatic Updates/Windows	Control Panel	All O/S
Backup and Restore Center	Control Panel	W7, W8, W10
BitLocker Enable	Control Panel	W7, W8, W10
Calibrate Color	Control Panel	Server 2008, W7, W8, Server 2012, W10
Certificate Manager	Management Snapin	All O/S
Clear Type Text	Control Panel	W7, W8, W10
Component Services	Management Snapin	All O/S
Computer Management	Management Snapin	All O/S
Data Sources	Control Panel	All O/S
Date and Time	Control Panel	All O/S
Default Location	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012
Defragment	Management Snapin	All O/S
Desktop DPI	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012
Device Manager	Management Snapin	All O/S
Disk Management	Management Snapin	Windows Server 2008, W7, W8, Windows Server 2012, W10

Component Name	Type	Operating System
Display	Control Panel	All O/S
Ease of Access Center	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10
Easy Transfer	Control Panel	W7, W8
Event Viewer	Management Snapin	All O/S
Explorer <ul style="list-style-type: none"> <li>• Advanced Security, Change Permissions</li> <li>• Show Quota Setting</li> <li>• Advanced Sharing</li> <li>• Check Drive for Errors</li> <li>• Security, Edit Users or Groups</li> </ul>	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10
Indexing Options <ul style="list-style-type: none"> <li>• Advanced Indexing Options</li> <li>• Pause Indexing</li> <li>• Show All Locations</li> </ul>	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10
Install/Uninstall Languages	Management Snapin	Windows Server 2008, W7, W8, Windows Server 2012, W10
Internet Options	Control Panel	All O/S
iSCSI Initiator	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10
Local Security Policy	Management Snapin	Windows Server 2008, W7, W8, Windows Server 2012, W10
Network Connections <ul style="list-style-type: none"> <li>• Advanced Sharing Settings</li> <li>• Local Area Connection Properties</li> </ul>	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10

Component Name	Type	Operating System
<ul style="list-style-type: none"> <li>• Manage Local Area Connection (Disable/Delete/Rename/Bridge)</li> <li>• Manage Wireless Profile</li> <li>• Wireless Network Properties, Copy Profile to USB Flash Drive</li> <li>• Wireless Network Properties, Show Characters</li> </ul>		
Offline Files	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10
Performance Monitor	Management Snapin	All O/S
Power Options	Control Panel	All O/S
Printers	Control Panel	All O/S
Problem Reporting	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10
Recovery Disc	Control Panel	W7, W8, W10
Recovery Restore	Control Panel	W7, W8, W10
Regional Options	Control Panel	All O/S
Server Manager	Management Snapin	Windows Server 2008, Windows Server 2012
Services	Management Snapin	All O/S
System	Control Panel	All O/S
System Configuration	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10
Task Scheduler	Management Snapin	Windows Server 2008, W7, W8, Windows Server 2012, W10
Troubleshoot	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012,



Component Name	Type	Operating System
		W10
Trusted Platform	Management Snapin	W7, W8, W10
User Accounts <ul style="list-style-type: none"><li>• Manage User Accounts</li><li>• User Account Control Settings</li></ul>	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10
Windows Defender	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10
Windows Features	Control Panel	W7, W8, W10
Windows Firewall Advanced Settings	Management Snapin	Windows Server 2008, W7, W8, Windows Server 2012, W10
Windows Firewall Settings <ul style="list-style-type: none"><li>• Action Center Firewall Settings</li><li>• Control Panel Firewall Settings</li></ul>	Control Panel	Windows Server 2008, W7, W8, Windows Server 2012, W10