SCCM Client Actions Tool PowerShell Edition aka PoshCAT

|  |
| --- |
| Coretech_brevpapir |

|  |  |
| --- | --- |
| Author: | Kaido Järvemets |
| Create date: | 08-08-2013 |
| Change date: | 28-08-2013 |
| Document version no.: | 0.2 |

Contents

[Coretech License terms for using the product 3](#_Toc365462915)

[Description 5](#_Toc365462916)

[Features 5](#_Toc365462917)

[Tool Actions 6](#_Toc365462918)

[Client schedule actions 6](#_Toc365462919)

[Client agent actions 6](#_Toc365462920)

[Client health actions 6](#_Toc365462921)

[Software Updates and Endpoint Protection Actions 6](#_Toc365462922)

[Application Virtualization 7](#_Toc365462923)

[Deployment / Advertisement Actions 7](#_Toc365462924)

[Other actions 7](#_Toc365462925)

[PoshCAT requirements 7](#_Toc365462926)

[PoshCAT Installation 8](#_Toc365462927)

[Enabling PowerShell Remoting through Group Policy for PoshCAT 10](#_Toc365462928)

[Testing remote commands with PowerShell Console 11](#_Toc365462929)

[PoshCAT functionality 12](#_Toc365462930)

[Configuration File 12](#_Toc365462931)

[Logging 12](#_Toc365462932)

[Specifying computer list 12](#_Toc365462933)

[Add Computer prompt 12](#_Toc365462934)

[Configuration Manager Collection Import 13](#_Toc365462935)

[CSV file import 13](#_Toc365462936)

[TXT file import 14](#_Toc365462937)

[Active Directory Connector 14](#_Toc365462938)

[Reloading Configuration File 14](#_Toc365462939)

[PoshCAT functions 15](#_Toc365462940)

[Client agent actions 15](#_Toc365462941)

[Client health actions 16](#_Toc365462942)

[Software Updates and Endpoint Protection Actions 17](#_Toc365462943)

[Application Virtualization 17](#_Toc365462944)

[Other actions 18](#_Toc365462945)

# Coretech License terms for using the product

**CORETECH A/S SOFTWARE LICENSE TERMS**

These license terms are an agreement between Coretech A/S and you. Please read them. They apply to the software you are downloading from Coretech.dk, which includes the media on which you received it, if any. The terms also apply to any Coretech A/S updates for this software, unless other terms accompany those items. If so, those terms apply.

**BY USING THE SOFTWARE, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO NOT USE THE SOFTWARE.**

**If you comply with these license terms, you have the rights below.**

**1. INSTALLATION AND USE RIGHTS.** You may install and use any number of copies of the software on your devices.

**2. Scope of License.** The software is licensed, not sold. This agreement only gives you some rights to use the software. Coretech reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the software only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the software that only allow you to use it in certain ways. You may not

· Work around any technical limitations in the binary versions of the software;

· reverse engineer, decompile or disassemble the binary versions of the software, except and only to the extent that applicable law expressly permits, despite this limitation;

· make more copies of the software than specified in this agreement or allowed by applicable law, despite this limitation;

· publish the software for others to copy;

· rent, lease or lend the software;

· transfer the software or this agreement to any third party; or

**3. DOCUMENTATION.** Any person that has valid access to your computer or internal network may copy and use the documentation for your internal, reference purposes.

**4. SUPPORT SERVICES.** Because this software is "as is," we may not provide support services for it.

**5. Entire Agreement.** This agreement, and the terms for supplements and updates that you use, are the entire agreement for the software and support services.

**6. Applicable Law.**

**a. United States.** If you acquired the software in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.

**b. Outside the United States.** If you acquired the software in any other country, the laws of that country apply.

**7. Legal Effect.** This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the software. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.

**8. Disclaimer of Warranty. The software is licensed "as-is." You bear the risk of using it. CORETECH A/S gives no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this agreement cannot change. To the extent permitted under your local laws, CORETECH A/S excludes the implied warranties of merchantability, fitness for a particular purpose and non-infringement.**

**9. Limitation on and Exclusion of Remedies and Damages. You can recover from CORETECH and its suppliers only direct damages up to U.S. $1.00. You cannot recover any other damages, including consequential, lost profits, special, indirect or incidental damages.**

This limitation applies to

· Anything related to the software, services, content (including code) on third party Internet sites, or third party programs; and

· claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

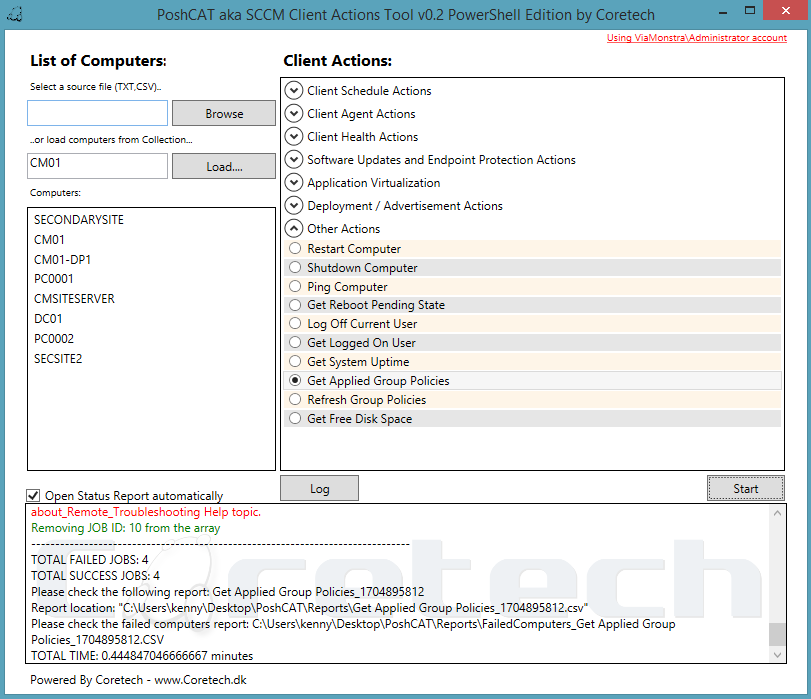
It also applies even if Coretech A/S knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Description   
**SCCM Client Actions Tool PowerShell Edition aka** PoshCAT is a practical and simple PowerShell application for performing most common day-to-day administrative tasks on System Center 2012 Configuration Manager Clients. The tool allows running actions remotely on one or more computers simultaneously. A list of computers can be provided either from a file (CSV, TXT) or loaded from Configuration Manager Collection importer or through Add Computer prompt.

You can find the old version from here **SCCMCAT** - <https://sccmcat.codeplex.com/> by Christjan Schumann.

## Features

* Multithreaded – 10 jobs at one time
* CSV reporting
* Configuration Manager 2012 Collection browser
* Configuration file for Client Actions
* Possibility to add own functions
* Support for two different job types
* Ability to create different Client Actions sets for different Support groups
* Alternate Credentials Support
* Active Directory Connector



Currently the tasks that can be performed are classified by 7 categories:

# Tool Actions

## Client schedule actions

* Hardware Inventory Cycle
* Software Inventory Cycle
* Discovery Data Collection Cycle (Send DDR)
* Machine Policy Retrieval & Evaluation Cycle
* Software Updates Deployment Evaluation Cycle
* Software Updates Scan Cycle
* File Collection Cycle
* Windows Installer Source List Update Cycle
* Software Metering Usage Report Cycle
* Desired Configuration Manager Cycle
* Application Machine Policy Cycle
* Application Global Evaluation Cycle

## Client agent actions

* Get ConfigMgr Client Management Point
* Restart SMS Agent Host service
* Get SMS Agent Host Service State
* Reassign ConfigMgr Client Site Code
* Create a new SMSCFG.ini file
* Reset ConfigMgr Client policies
* Repair ConfigMgr Client
* Uninstall ConfigMgr Client
* Install ConfigMgr Client
* Get ConfigMgr Client Inventory Dates
* Get ConfigMgr Client Cache Information
* Remove ConfigMgr Client Cache Items
* Set ConfigMgr Client Cache Size

## Client health actions

* Reset WMI Repository
* Salvage WMI Repository
* Get WMI Repository State
* Get Admin$ Share Status

## Software Updates and Endpoint Protection Actions

* Get ConfigMgr Client WSUS Content Location
* Get Windows Update Agent Version
* Get Missing Updates
* Install Missing Updates
* Query Updates Status
* Start Endpoint Protection Full Scan
* Get Endpoint Protection Applied Policies
* Get Endpoint Protection Last Scan Date

## Application Virtualization

* Get APP-V Client Packages
* Get APP-V Client Configuration
* Get APP-V Client Version
* Enable APP-V Client Package Scripts

## Deployment / Advertisement Actions

* Get ConfigMgr Client Available Applications

## Other actions

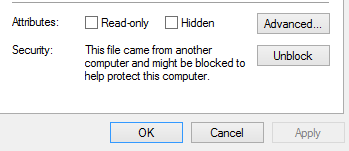
* Restart Computer
* Shut down Computer
* Ping Computer
* Get Reboot Pending State
* Log off current user
* Get logged-on user
* Get system uptime
* Get Applied Group Policies
* Refresh Group Policies
* Get Free Disk Space

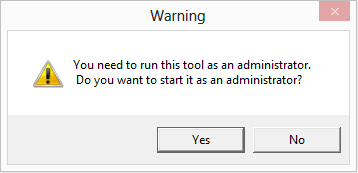
# PoshCAT requirements

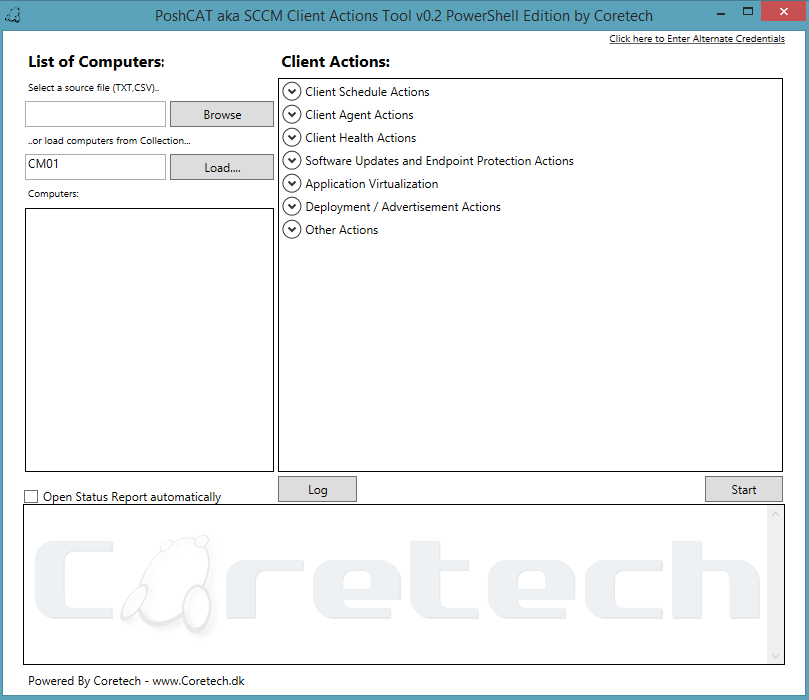
* Windows Management Framework 3.0
* Microsoft .Net Framework 4.0 or newer
* Windows 7, Windows 8 or Windows Server 2012
* Administrative Permissions
* PowerShell Remoting

# PoshCAT Installation

1. Download and Unzip **PoshCAT\_v0.1.zip**
2. Right-Click **Start-PoshCAT.ps1**, **WorkerFunctions.ps1**, **SharedFunctions.ps1** and **Commands.xml** and select **properties**.
3. **Unblock** the files if needed and click OK.

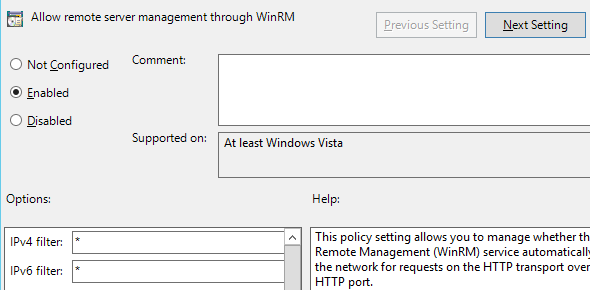


1. Right-click **Start-PoshCAT.ps1** and choose **Run with PowerShell**
2. If you don’t have administrative permissions, this message will be shown.  
     
   

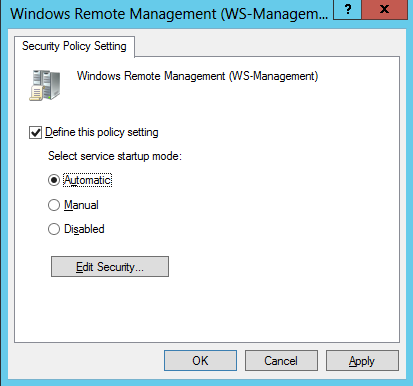


# Enabling PowerShell Remoting through Group Policy for PoshCAT

1. Open **Group Policy Management**
2. Create a new **Group Policy Object**, for example ***PowerShell Settings***
3. Open ***PowerShell Settings*** *GPO* with Group Policy Management Editor
4. Expand **Computer Configuration**; expand Policies; expand Administrative Templates; expand Windows Components; expand Windows Remote Management (WinRM)
5. Select **WinRM Service**
6. Open “**Allow remote server management through WinRM**” setting
7. Enable the Policy and set the **IPv4** and **IPv6** filter values to **\***



1. Click **OK**
2. Navigate to Windows Settings; expand Security Settings and select **System Services**
3. Select ***Windows Remote Management (WS-Management)*** Service and set the startup mode to Automatic



1. Click OK
2. Close the **Group Policy Editor**
3. Link the ***PowerShell Settings*** GPO to correct OU for testing
4. Reboot test computers

*Note: You can configure Windows Services through Group Policy Preferences also and if Firewall is also enabled in your environment, then you need to configure Firewall exceptions also.*

## Testing remote commands with PowerShell Console

Now log in to another machine and execute the following commands through PowerShell console or PowerShell ISE against the test machine(s).

#TEST 1

Get-WmiObject -Class Win32\_Share -ComputerName PC0002

#TEST 2

Get-Service -Name Winmgmt -ComputerName PC0002

#TEST 3

Invoke-Command -ScriptBlock {Get-Service -Name Winmgmt} -ComputerName PC0002

If these commands does not fail, then your Group Policy is configured correctly.

# PoshCAT functionality

## Configuration File

*PoshCAT* uses **XML** file for Client Actions and default settings. During the startup *PoshCAT* reads all the Client Actions and default settings from **commands.xml** file. *ToolActions* node describes all the available actions. By default there are over 30 client actions.

* **TASK** – Client Action name in UI



* **Component** – *PoshCAT* uses this property to group different Client Actions
* **ScriptBlock** – function name for specific task
* **Report** – this property can be *TRUE*/*FALSE*. If the value is *TRUE*, then the tool creates automatically *CSV* report. This property only works, if the function is configured correctly to return *PSObject*
* **JobType** – *PoshCAT* supports two different job types – *LOCAL* or *REMOTE*. If the *JobType* property is *LOCAL,* then it uses **Start-Job** cmdlet, for example commands like *Ping Computer*, *Restart Computer* etc. and if there is no *JobType* property, then it uses **Invoke-Command** to execute the command on remote computer.



*Example from commands.xml*

* **Parameter** – *PoshCAT* uses this property as a function parameter, for example actions like *Hardware Inventory Cycle, Set ConfigMgr Client Cache Size* etc.



*Example from commands.xml*

## Logging

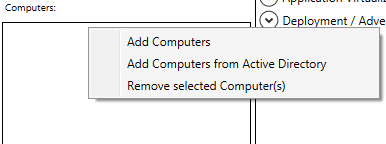
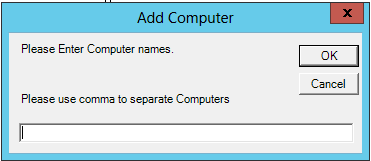
*PoshCAT* logs all activity to **PoshCAT.log**.Log file is saved in same folder as the tool. The log is formatted for viewing with *Configuration Manager Trace Log Tool* by default. Currently there is no log file size limit.

## Specifying computer list

*PoshCAT* can import computers from **TXT** and **CSV** file, from Configuration Manager Collection or you can use “*Add Computer*” prompt to add one or more computers.

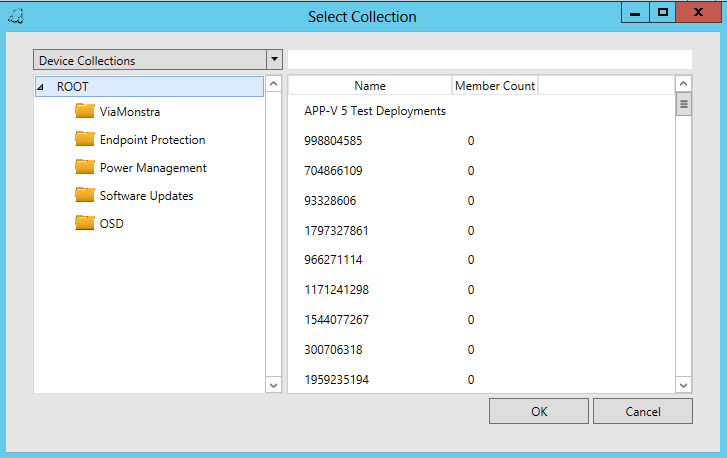
### Add Computer prompt

To add computers through ***Add Computer*** prompt, right-click on computers box and select ***Add Computer***

*Adding Computers through “****Add Computer****” prompt*

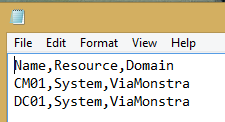
### Configuration Manager Collection Import



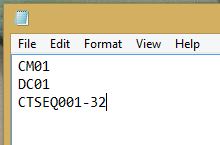
*Adding Computers through* ***Configuration Manager Collection***

### CSV file import

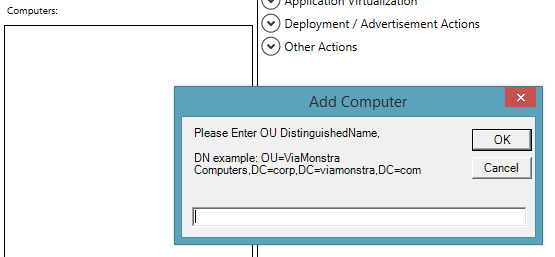
To import **CSV** file you need to make sure that the computer name column name starts always with **NAME**



### TXT file import



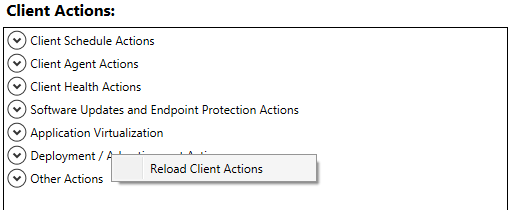
### Active Directory Connector



*Adding Computers through* ***Active Directory Connector***

## Reloading Configuration File

If the tool is already running and you have made some changes to **commands.xml** file, then you can easily import the configuration file again making right-click on ***Client Actions*** box and choose ***Reload Client Actions***



## PoshCAT functions

### Client agent actions

#### Get ConfigMgr Client Management Point

**FUNCTION:** Get-CMClientManagementPoint

This function queries Configuration Manager Client current Management Point and returns PSObject for reporting.

#### Restart SMS Agent Host service

**FUNCTION:** Restart-ConfigMgrClientService

This function restarts Configuration Manager Client SMS Agent Host Service and returns PSObject for reporting.

#### Get SMS Agent Host Service State

**FUNCTION:** Get-SMSAgentHostServiceState

This function queries Configuration Manager Client SMS Agent Host Service status and returns PSObject for reporting.

#### Reassign ConfigMgr Client Site Code

**FUNCTION:** Set-CMClientSiteCode

This function modifies Configuration Manager Client Site Code. Default Site Code **PS1** is configured in **commands.xml** file

#### Create a new SMSCFG.ini file

**FUNCTION:** Create-SMSCFGIniFile

This function stops SMS Agent Host service and renames **SMSCFG.ini** to **SMSCFG.ini.old** and starts again SMS Agent Host service

#### Reset ConfigMgr Client policies

**FUNCTION:** Reset-CMClientPolicy

This function resets Configuration Manager Client policies

#### Repair ConfigMgr Client

**FUNCTION:** Start-CMClientRepair

This function starts Configuration Manager Client repair

#### Uninstall ConfigMgr Client

**FUNCTION:** Remove-CMClient

This function uninstalls Configuration Manager Client

#### Install ConfigMgr Client

**FUNCTION:** Install-CMClient

This function installs Configuration Manager Client. This functions creates Scheduled task to remote computer that runs only once. ConfigMgr Client installation is scheduled to run under Local System account and you need to allow Domain Computers to read ConfigMgr Client installation folder.

#### Get ConfigMgr Client Inventory Dates

**FUNCTION:** Get-CMClientInventoryActions

This function queries Configuration Manager Client hardware and software inventory dates and returns PSObject for reporting

#### Get ConfigMgr Client Cache Information

**FUNCTION:** Get-CMClientCacheInformation

This function queries Configuration Manager Client Cache information and returns PSObject for reporting.

#### Remove ConfigMgr Client Cache Items

**FUNCTION:** Remove-CMClientCacheItems

This function removes Configuration Manager Client Cache items

#### Set ConfigMgr Client Cache Size

**FUNCTION:** Set-CMClientCacheSize

This function configures Configuration Manager Client Cache Size. Default Cache size is **10000** **MB** and you can configure the size in **Commands.xml** file.

### Client health actions

#### Reset WMI Repository

**FUNCTION:** Invoke-WMIStateCheck

This function stops Windows Management Instrumentation service, registers all the DLLs in WBEM folder and resets the WMI Repository using ***WinMgmt.exe*** tool.

#### Salvage WMI Repository

**FUNCTION:** Salvage-WMIRepository

Performs a consistency check on the WMI repository, and if an inconsistency is detected, rebuilds the repository. The content of the inconsistent repository is merged into the rebuilt repository, if it can be read. The salvage operation always works with the repository that the WMI service is currently using. MOF files that contain the #pragma autorecover preprocessor statement are restored to the repository. This function returns PSObject for reporting

#### Get WMI Repository State

**FUNCTION:** Get-WMIRepositoryState

Performs a consistency check on the WMI repository. When you add the /verifyrepository switch without the <path> argument, then the live repository currently used by WMI is verified. When you specify the path argument, you can verify any saved copy of the repository. In this case, the path argument should contain the full path to the saved repository copy. The saved repository should be a copy of the entire repository folder. This function returns PSObject for reporting

#### Get Admin$ Share Status

**FUNCTION:** Get-AdminShare

This function queries ADMIN$ share status and returns PSObject for reporting

### Software Updates and Endpoint Protection Actions

#### Get ConfigMgr Client WSUS Content Location

**FUNCTION:** Get-CMClientWSUSContentLocation

This function queries Configuration Manager Client WSUS content location and version. If the WMI query fails, then it failover to registry keys. This function returns PSObject for reporting

#### Get Windows Update Agent Version

**FUNCTION:** Get-CMClientWUAVersion

This function queries Configuration Manager Client Windows Update Agent version. If the WMI query fails, then it failover to wuapi.dll file. This function returns PSObject for reporting

#### Get Missing Updates

**FUNCTION:** Get-CMClientMissingUpdates

This function queries all the available updates for Configuration Manager Client and returns PSObject for reporting

#### Install Missing Updates

**FUNCTION:** Install-CMClientMissingUpdates

This function installs all available updates

#### Query Updates Status

**FUNCTION:** Get-WindowsUpdateStatus

This function queries specific update(s) installation status and returns PSObject for reporting

#### Start Endpoint Protection Full Scan

**FUNCTION:** Start-EPFullScan

This function starts Endpoint Protection Client full scan

#### Get Endpoint Protection Applied Policies

**FUNCTION:** Get-EPAppliedPolicies

This function queries Endpoint Protection Client applied policies and returns PSObject for reporting

#### Get Endpoint Protection Last Scan Date

**FUNCTION:** Get-EPlastScanTime

This function queries Endpoint Protection Client last scan date and returns PSObject for reporting

### Application Virtualization

#### Get APP-V Client Package

**FUNCTION:** Get-APPvClientPackages

This function queries App-V Client packages and returns PSObject for reporting

#### Get APP-V Client Configuration

**FUNCTION:** Get-AppVClientConfiguration

This function queries App-V Client configuration and returns PSObject for reporting

#### Get APP-V Client Version

**FUNCTION:** Get-AppvClientVersion

This function queries App-V Client version and installation location and returns PSObject for reporting

#### Enable APP-V Client Package Scripts

**FUNCTION:** Enable-AppVClientScripts

This function enables App-V Client Package scripts.

Deployment / Advertisement Actions

#### Get ConfigMgr Client Available Applications

**FUNCTION:** Get-ConfigMgrClientAvailableApps

This function queries ConfigMgr Client published applications and returns PSObject for reporting

### Other actions

#### Restart Computer

**FUNCTION:** Restart-CMClientComputer

This function restarts computer

#### Shut down Computer

**FUNCTION:** Start-ComputerShutDown

This function shutdowns computer

#### Ping Computer

**FUNCTION:** Ping-Computer

This function ping computer and returns PSObject for reporting

#### Get Reboot Pending State

**FUNCTION:** Get-RebootPendingStatus

This function queries computer reboot pending state and returns PSObject for reporting. If the WMI query fails, then it failover to registry

#### Log off current user

**FUNCTION:** Invoke-UserLogOff

This function logs off remote computer user

#### Get logged-on user

**FUNCTION:** Get-LoggedOnUser

This function queries currently logged on user and returns PSObject for reporting.

#### Get system uptime

**FUNCTION:** Get-ComputerUpTime

This function queries computer uptime and returns PSObject for reporting

#### Get Applied Group Policies

**FUNCTION:** Get-ComputerAppliedPolicies

This function queries computer applied Group Policies and returns PSObject for reporting

#### Refresh Group Policies

**FUNCTION:** Invoke-GPUpdate

This function invokes Group Policy update

#### Get Free Disk Space

**FUNCTION:** Get-FreeDiskSpace

This function queries computer free space for fixed drives and returns PSObject for reporting