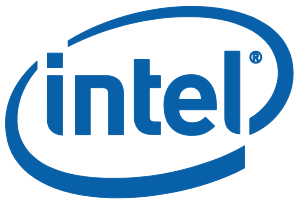


Intel vPro® Technology Module for Microsoft* Windows* PowerShell*

Installation and User Guide

Module Version 18.0.0.1
Document Updated: March 2024

Revision History

Module Version	Revision History	Date
1.0	Initial release	August, 2010
2.0	Document additions and changes to support version 2.0 of the PowerShell Module for Intel vPro	October, 2010
3.0	Document additions and changes to support version 3.0 of the Windows PowerShell Module for Intel vPro Technology	March, 2011
3.1	Document additions and changes to support version 3.1 of the Windows PowerShell Module for Intel vPro Technology	June, 2011
3.2	Document additions and changes to support version 3.2 of the Windows PowerShell Module for Intel vPro Technology	Dec, 2011
3.2.2	Document additions and changes to support version 3.2.2 of the Windows PowerShell Module for Intel vPro Technology	Mar, 2012
3.2.4	Document additions and changes to support version 3.2.4 of the Windows PowerShell Module for Intel vPro Technology	Feb, 2013
3.2.5	Document additions and changes to support version 3.2.5 of the Windows PowerShell Module for Intel vPro Technology	Aug, 2013
3.2.6	Document additions and changes to support version 3.2.6 of the Windows PowerShell Module for Intel vPro Technology	Jan, 2014
3.2.7	Document additions and changes to support version 3.2.7 of the Windows PowerShell Module for Intel vPro Technology. Code examples for implementing the Write-AmtCredential and Read-AmtCredential cmdlets, which were removed from the module in this release.	Nov, 2019
3.2.7	Document update: System Requirements and Importing the Module	September 2020
3.2.7	Document update: Updated Close IDER and Close SOL	January 2021
15.0.2.1 (numbering system changed)	Document update: Added: • TLS Configuration Flow Using PowerShell Snippets, including cmdlets • One Click Recovery cmdlet	July 2021
15.0.3.1	Document update: • Added "Adding a new script" • Removed the -EnableLocalTLS option from the Invoke-ConfigureTLSMutualAuthentication cmdlet	September 2021
15.0.3.1	Updated Invoke-AMTPowerManagement options and Get-AMTPowerState output example	October 2021
16.0.4.1	Added documentation for Get-UniquePlatformIDFeatureInfo command	June 2022
16.0.4.1	Added -AcceptSelfSignedCert option to relevant cmdlets.	July 2022
18.0.0.1	Added support for Microsoft* Windows* PowerShell* 7.0	March 2024

Contents

Figures

Tables

1 Preface

Microsoft* Windows* PowerShell* is becoming more prevalent as an automation scripting language within many Information Technology (IT) environments. Whether writing scripts to automate tasks or taking advantage of native Windows PowerShell extensibility within existing management tools, the ability to Out of Band manage Intel® Active Management Technology (Intel® AMT) enabled clients with Windows PowerShell is a very attractive solution.

1.1 Document Scope

This document covers the requirements, installation and usage of the Windows PowerShell Module for Intel vPro® Technology.

1.2 Intended Audience

Windows PowerShell command line shell and scripting language helps IT professionals achieve greater control and productivity. Using a new administrator-focused scripting language and consistent syntax and utilities Windows PowerShell allows IT professionals to more easily control system administration and accelerate automation. This document is intended for IT professionals who desire to learn more about using the Intel® vPro™ Module for Windows PowerShell.

1.3 Related Documentation and Software

The download package and supporting files referenced in this document can be found at the following links:

https://software.intel.com/en-us/download/intel-active-management-technology-sdk

http://www.intel.com/go/powershell

Microsoft Windows Management Framework (which includes Windows PowerShell):
http://support.microsoft.com/kb/968929

Microsoft Windows Remote Manager (WinRM):
http://www.microsoft.com/downloads/details.aspx?familyid=845289ca-16cc-4c73-8934-dd46b5ed1d33&displaylang=en

2 Introduction

The Windows PowerShell command line shell and scripting language helps IT professionals achieve greater control and productivity. Using a new administrator focused scripting language and consistent syntax and utilities, Windows PowerShell allows IT professionals to more easily control system administration and accelerate automation. Windows PowerShell is easy to adopt, learn, and use. It works with existing IT infrastructure and cmdlet investments. It runs on Windows XP, Windows Vista*, Windows Server* 2003 and is included as part of Windows 7, Windows Server 2008, Windows Server 2008 R2 and later. For more information on Windows PowerShell), please visit:

https://docs.microsoft.com/en-us/powershell/

By leveraging the Out of Band Management cmdlets within the Windows PowerShell Module for Intel vPro Technology, IT professionals can extend their PowerShell reach to include direct manageability of Intel AMT enabled clients independent of power or operating system health.

2.1 Requirements

IT / Console PC	Any PC with Microsoft Windows and: • Windows PowerShell versions up to and including 7.0 • Windows Remote Management (WinRM)
Managed Client with Intel vPro technology	• Intel® Active Management Technology (Intel® AMT) 3.0 or higher. • Intel® Management Engine is provisioned. • See subsections 2.1.1, 2.1.2, and 2.1.3 for further client requirements.

2.1.1 Setup and Configuration of the Intel vPro® Technology Based Client Prior to Module Use

Prior to using the Intel® vPro™ Technology module for Windows PowerShell, the client's Intel AMT firmware must be set up and configured. Use existing Configuration Management software or reference the links below on how to set up and configure an Intel AMT enabled client.

- Intel® AMT Implementation and Reference Guide
- Intel® AMT SDK
- Intel® vPro™ Platform Forum

The Intel vPro PowerShell module can be used to set up and configure an Intel AMT enabled client.

See section 6.10.2 for information on the **Enter-AMTRemoteConfiguration** cmdlet.

2.1.2 Cmdlet and Function Authentication

Credentials must be specified in order to invoke commands against the Intel vPro technology enabled client. Typical behavior of the Windows PowerShell Module for Intel vPro technology cmdlets and functions is as follows:

- When no credentials are provided, the cmdlets and functions use the locally logged-on Kerberos credential.
- When only the username (Kerberos or Digest) parameter is included, a prompt is displayed to provide the associated password.
- If the credentials are stored as a PowerShell variable, they may be passed into the cmdlets and functions with the credential parameter.

NOTE

For Active Directory authentication to work correctly, a hostname or the Fully Qualified Domain Name (FQDN) must be specified.

2.1.3 Cmdlet and Function Communication Encryption

If the Intel vPro technology enabled client is configured to use Transport Layer Security (TLS) by having a web server certificate issued to the Intel Management Engine, the **–TLS** switch must be passed to the cmdlet.

When managing an Intel vPro technology enabled client over TLS (port 16993), it is important that the computer name match the primary subject name of the issued TLS certificate. This is typically the Fully Qualified Domain Name (FQDN).

2.2 Configuration and Usage Process Overview

The Windows Powershell configuration and usage process consists of three primary steps:

1. Install Windows PowerShell if it has not yet been installed.
2. Install the Windows PowerShell Module for Intel vPro Technology.
3. Use the Intel vPro PowerShell cmdlets.

3 Windows* PowerShell Setup and Configuration

This chapter and its subsections describe how to:

- Set up and configure the console PC to use Windows PowerShell
- Install the Windows PowerShell Module for Intel vPro technology
- Import the module once it has been installed

3.1 Installing Windows* PowerShell

Windows PowerShell is natively included with Windows Server 2008, Windows Server 2008 R2, Windows 7 and above.

Windows Management Framework makes some updated management functionality in Windows 7 and in Windows Server 2008 R2 available to be installed on Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008. Windows Management Framework contains Windows Remote Management (WinRM) 2.0, Windows PowerShell 2.0, and Background Intelligent Transfer Service (BITS) 4.0.

To obtain a copy of the Windows Management Framework and install Windows PowerShell, click the link below.

http://support.microsoft.com/kb/968929

3.1.1 Configuring Windows PowerShell

By default, Windows PowerShell has its ExecutionPolicy set to **Restricted**. This setting must be changed to execute the Out of Band Management PowerShell cmdlets and functions provided within the PowerShell Module for Intel vPro technology.

All the cmdlets and functions within the PowerShell Module for Intel vPro technology have been signed. At a minimum, the PowerShell Execution Policy needs to be changed to **RemoteSigned** to execute the cmdlets and functions. If there are more restrictive security requirements, set the ExecutionPolicy to **AllSigned**.

Note that the AllSigned policy also allows running scripts that were changed on this computer.

To apply the ExecutionPolicy to the LocalMachine, run the following command within the Windows PowerShell Console (be sure to start the console with "Run as administrator"):

Set-ExecutionPolicy RemoteSigned

Or

Set-ExecutionPolicy –Scope LocalMachine RemoteSigned

To apply the ExecutionPolicy to the current user only, run the following command within the Windows PowerShell Console:

Set-ExecutionPolicy –Scope CurrentUser RemoteSigned

To apply the ExecutionPolicy to the process only, run the following command within the Windows PowerShell Console:

Set-ExecutionPolicy –Scope Process RemoteSigned

NOTE

If using an ExecutionPolicy based process, it will be required to run Set-ExecutionPolicy each time a Windows PowerShell Console is launched.

For more information on setting the Windows PowerShell ExecutionPolicy, please visit the following site:

http://msdn.microsoft.com/en-us/library/bb648601(VS.85).aspx

3.2 Installing the Windows PowerShell Module for Intel vPro Technology

This section describes how to install the Windows PowerShell Module for Intel vPro technology.

3.2.1 Downloading the Module

Download the latest copy of the Windows PowerShell Module for Intel vPro technology from the following location:

http://www.intel.com/go/powershell

3.2.2 Installing the Module

1. Uninstall previous versions of the Windows PowerShell Module for Intel vPro technology.

   NOTE

   Previous releases were installed using an installer.
   This release (3.2.7) includes a zip folder only.

2. Decompress the zip file to a directory.

3. Navigate to the directory where the file was decompressed.

4. Navigate to the \bin folder.
5. From the \bin folder, open PowerShell as Administrator.
6. Run the script: **Update-PSModulePath.ps1**. This script updates the **PSModulePath** environment variable with the new path of the IntelvPro Module. You can now run the module commands from any location, and not just from the module folder.

3.3 Configuring a Profile for the Windows PowerShell Module for Intel vPro Technology

Microsoft states that "A well-designed profile can make it even easier to use Windows PowerShell and to administer your system". This holds true for administering Intel vPro technology enabled devices. A well-designed PowerShell profile can make that task even easier.

Please view the link below from Microsoft for more information about PowerShell profiles:

http://msdn.microsoft.com/en-us/library/bb613488(v=vs.85).aspx

3.3.1 Setting Up a Profile for Intel vPro Technology

Below is an example of a profile you can put in %my documents%/WindowsPowerShell/Microsoft.PowerShell_profile.ps1.

```
function vPro
{
    Import-Module IntelvPro
}
```

Once you have created this profile, you can type **vPro** from within PowerShell to load the module.

3.3.2 Using Intel® AMT Credential Secure Storage

Intel AMT credentials can be securely stored in a PowerShell encrypted string by using a cmdlet as described below.

Note that any implementation of this cmdlet should ensure that the credentials saved in the file are protected from malicious usage.

NOTE

In Release 3.2.7 of the module, the code has been removed for reasons of security. Instead, we are providing the text of the cmdlet so you can insert it yourself, after taking the necessary precautions.

The following cmdlet code allows you to save the Intel AMT credentials (contained in the **$AmtCred** variable) in a file on the computer. This allows the privileged administrator to store the Intel AMT credential without its being exposed in plain text for any user to view. The cmdlet uses secure strings to do this. You need to take the necessary precautions to make sure that an unauthorized user does not decrypt the credentials, since this will allow them to access Intel AMT functionality.

A key parameter can be passed in to additionally encrypt the password for maximum security.

A **FilePath** parameter may be used to explicitly specify a file in which to store the Intel AMT Credential. The credential file may then be moved between systems. It is strongly recommended to encrypt the file additionally using a key.

Write-AmtCredential Function Code

```
Function Write-AmtCredential {
<#
  .Synopsis
    Writes an Intel Active Management Technology credential from secure string
storage
  .Description
    Writes an Intel Active Management Technology(AMT)credential to
System.Security.SecureString in the default user path.
  .Link
    http:\\vproexpert.com
    http:\\www.intel.com\vpro
    http:\\www.intel.com
  .Example
    Write-AmtCredential
  .Example
    $AMTCredential = Write-AmtCredential  (will assume the digest account
"admin")
  .Example
    $AMTCredential = Get-Credential
    Write-AmtCredential –Username $AMTCredential.Username –Password
$AMTCredential.Password
  .Example
    Write-AMTCredential [[-FilePath] <String>] [[-Key] <String>] [[-Hint]
<String>] [[-AsPlainText]] [[-Force]] [[-Username] <String>] [-Password]
<SecureString> [<CommonParameters>]
    #>


[CmdletBinding()]
Param (

[Parameter(Mandatory=$false,ValueFromPipelineByPropertyName=$true,ValueFromPipe
line=$true, position=0, HelpMessage="Path to Credential File")] [string]
$FilePath,

[Parameter(Mandatory=$false,ValueFromPipelineByPropertyName=$true,ValueFromPipe
line=$false, position=1, HelpMessage="An ASCII Key of 128,196 of 256 Length")]
[string] $Key,

[Parameter(Mandatory=$false,ValueFromPipelineByPropertyName=$true,ValueFromPipe
line=$false, position=2, HelpMessage="Password Hint")] [string] $Hint,

[Parameter(Mandatory=$false,ValueFromPipelineByPropertyName=$true,ValueFromPipe
line=$false, position=3, HelpMessage="Save password as plain text")]
[System.Management.Automation.SwitchParameter] $AsPlainText,

[Parameter(Mandatory=$false,ValueFromPipelineByPropertyName=$true,ValueFromPipe
line=$false, position=3, HelpMessage="Force")]
[System.Management.Automation.SwitchParameter] $Force,
```

```
[Parameter(Mandatory=$false,ValueFromPipelineByPropertyName=$true,ValueFromPipe
line=$false, position=4, HelpMessage="Amt user")] [string] $Username,

[Parameter(Mandatory=$true,ValueFromPipelineByPropertyName=$true,ValueFromPipel
ine=$false, position=5, HelpMessage="Amt Password")]
[System.Security.SecureString] $Password
 )


PROCESS {

    if ([string]::IsNullOrEmpty($Username))
    {
        $Username = "admin"
    }

    if ([string]::IsNullOrEmpty($FilePath))
    {
        $FilePath = Get-AmtCredentialPath
    }

    if ( (test-path $FilePath) -eq $false)
    {
        $fileItem = new-item -path $FilePath -itemtype file -force
    }


[Xml]$xml="<Credential><User>admin</User><Hint></Hint><Password></Password><Enc
ryption>DAPI</Encryption></Credential>"
    $xml.Credential.User = $Username

    if ([string]::IsNullOrEmpty($Hint) -eq $false)
    {
        $xml.Credential.Hint=$Hint
    }

    if ($AsPlainText.IsPresent)
    {
        if ($Force.IsPresent -eq $False)
        {
            throw (new-object
System.Management.Automation.PSInvalidOperationException  -ArgumentList "Saving
Plain Text Passwords requries the force switch")
        }
        $ptr= [Runtime.InteropServices.Marshal]::SecureStringToBSTR($Password)
        $pwdText = [Runtime.InteropServices.Marshal]::PtrToStringAuto($ptr)
```

```
            [Runtime.InteropServices.Marshal]::ZeroFreeBSTR($ptr)
            $xml.Credential.Password=$pwdText
            $xml.Credential.Encryption="PlainText"
            Set-Content $FilePath $xml.OuterXml
        }
        elseif ([string]::IsNullOrEmpty($Key))
        {
            $xml.Credential.Password= (ConvertFrom-SecureString -SecureString
  $Password).ToString()
            $xml.Credential.Encryption="DPAPI"
            Set-Content $FilePath $xml.OuterXml
        }
        else
        {
            [byte[]]$KeyBytes = [System.Text.Encoding]::ASCII.GetBytes($Key)
            $xml.Credential.Password= (ConvertFrom-SecureString -SecureString
  $Password -Key $KeyBytes).ToString()
            $xml.Credential.Encryption="Rijndael"
            Set-Content $FilePath $xml.OuterXml
        }
      }
  }
```

This cmdlet allows the privileged administrator to store the required Intel AMT credentials without the credentials being exposed in plain text for any user to view.

Once the credentials are have been stored, a subsequent PowerShell session can read them without exposing them, as in the snippet below:

Read-AmtCredential Function Code

```
Function Read-AmtCredential {
<#
  .Synopsis
    Reads an Intel Active Management Technology credential from secure string
storage
  .Description
    Reads an Intel Active Management Technology(AMT) credential from secure
string storage.
  .Notes
    Reads System.Security.SecureString from the default user path.
  .Link
    http:\\vproexpert.com
    http:\\www.intel.com\vpro
    http:\\www.intel.com
```

```
  .Example
    Read-AmtCredential
  .Example
    Read-AmtCredential [[-FilePath] <String>] [[-Key] <String>]
[<CommonParameters>]
  .Example
    $AMTCredential = read-AmtCredential


    #>
[CmdletBinding()]
Param (

[Parameter(Mandatory=$false,ValueFromPipelineByPropertyName=$true,ValueFromPipeli
ne=$true, position=0, HelpMessage="Path to Credential File")] [string] $FilePath,

[Parameter(Mandatory=$false,ValueFromPipelineByPropertyName=$true,ValueFromPipeli
ne=$false, position=1, HelpMessage="Encryption Key")] [String] $Key
 )



PROCESS {

    if ([string]::IsNullOrEmpty($FilePath))
    {
        $FilePath = Get-AmtCredentialPath
    }

    [xml]$xml=Get-Content $FilePath



    if ($xml.Credential.Encryption -eq "PlainText")
    {
        $password=ConvertTo-SecureString -AsPlainText -String
$xml.Credential.Password -Force
    }
    elseif ($xml.Credential.Encryption -eq "DPAPI")
    {
        $password=ConvertTo-SecureString -String $xml.Credential.Password
```

```
    }
    elseif ($xml.Credential.Encryption -eq "Rijndael")
    {
        [byte[]]$KeyBytes = [System.Text.Encoding]::ASCII.GetBytes($Key)
        $password=ConvertTo-SecureString -Key $KeyBytes -String
$xml.Credential.Password
    }
    new-object 'System.Management.Automation.PSCredential'  -ArgumentList
$xml.Credential.User,$password


 }
}
```

For testing purposes, the **–force** switch can be used to store plain text credentials.

To set your profile to load the module and set the Intel AMT credentials when you type **vPro** in a PowerShell session, change your profile as follows:

```
function vPro
{
    Import-Module IntelvPro
    New-Variable -Name AmtCred -Value (Read-AmtCredential)
}
```

If you want the WriteCredentials or ReadCredentials cmdlets to be part of the IntelvPro module, add them to the list in the IntelvPro.PSM1 file.

3.3.3 Making Everything Load Automatically

To make the module load and set the $AmtCred variable:

1. Use **Write-AMTCredential** to store AMTCred. See Write-AmtCredential Function Code on Page 8 for the code of this function.
2. Implement the **Read-AmtCredential** code to read the stored encrypted credentials. See Read-AmtCredential Function Code on Page 10 for the code of this function.
3. Each time a PowerShell session is started, modify the profile to include the following (not in a function block):

```
Import-Module IntelvPro
New-Variable -Name AmtCred -Value (Read-AmtCredential)
```

Note that the user is responsible for the encryption of the stored credentials.

3.3.4 Easily Mounting an AMTSystem PowerShell Drive

To easily mount an AMTSystem Powershell Drive, add the following function to the profile:

```
function mount-AMTDrive
{
  Param([string]$HostName,
[System.Management.Automation.PSCredential]$AMTCredential)
    process{
      New-PSdrive -scope global -name $HostName -psprovider amtsystem
        -root \ -computername $HostName -credential $AMTCredential
    }
}
```

Next mount an AMTSystem Powershell drive by typing:

Mount-AMTDrive $HostName

The drive name will be $HostName. To list the drive contents, type:

PSDrive

NOTE

The New-PSDrive cmdlet does not accept ~ / \ . : characters. It is recommended to use the Hostname instead of an IP address.

4 Using the Windows PowerShell Module for Intel vPro Technology

The Windows PowerShell console provides access to multiple modules at any given time. Before using a module's cmdlets and functions, the module must be imported. The module must be imported into each shell in which the module will be used.

4.1 Importing the Module

Run the **Import-Module** command to use a module's cmdlets and functions. To list the available modules, type **Get-Module –ListAvailable** within Windows PowerShell. If the Windows PowerShell Module for Intel® vPro™ Technology is installed as described in section 3.2, **IntelvPro** will be listed as one of the available modules.

To import the Windows PowerShell Module for Intel vPro Technology, type **Import-Module IntelvPro**, as shown below.

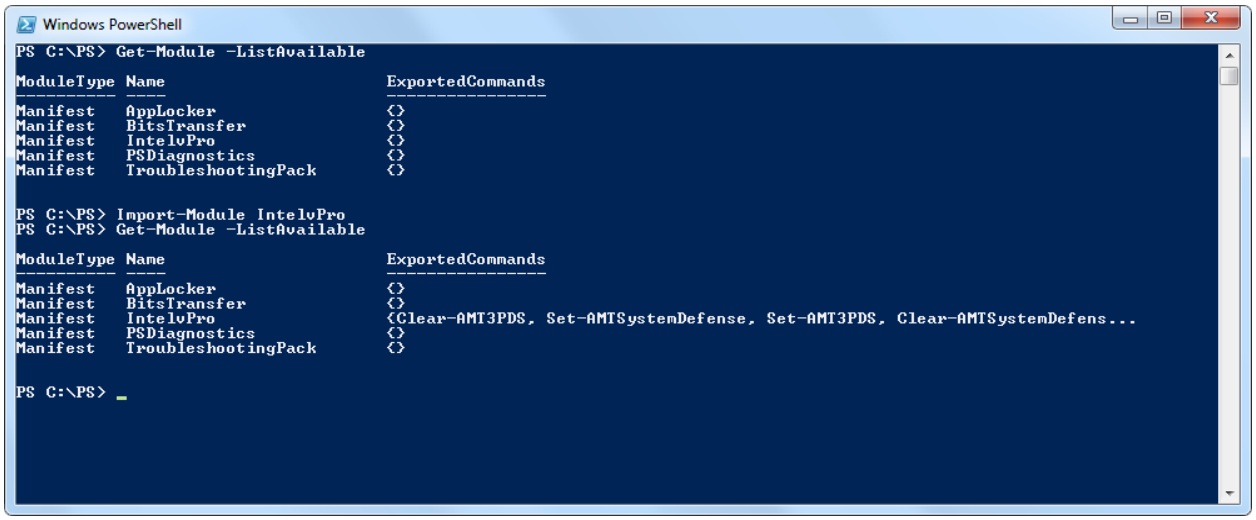


Figure 1: Importing the Module

NOTE

If **Import-Module** fails, launch PowerShell again, this time without Administrator privileges. The failure is caused by a PowerShell bug (the PowerShell EnvVariable is not cleared from the previous session).

After importing the module, type **Get-Module –ListAvailable** to verify that the module has been imported along with the available Exported Commands.

To load the module automatically when Windows PowerShell is started, add **Import-Module IntelvPro** to the Windows PowerShell Profile.ps1 file.

Once the module has been imported, its cmdlets can be listed by using the **Get-Command –Module IntelvPro** command, as shown in Figure 2 below.

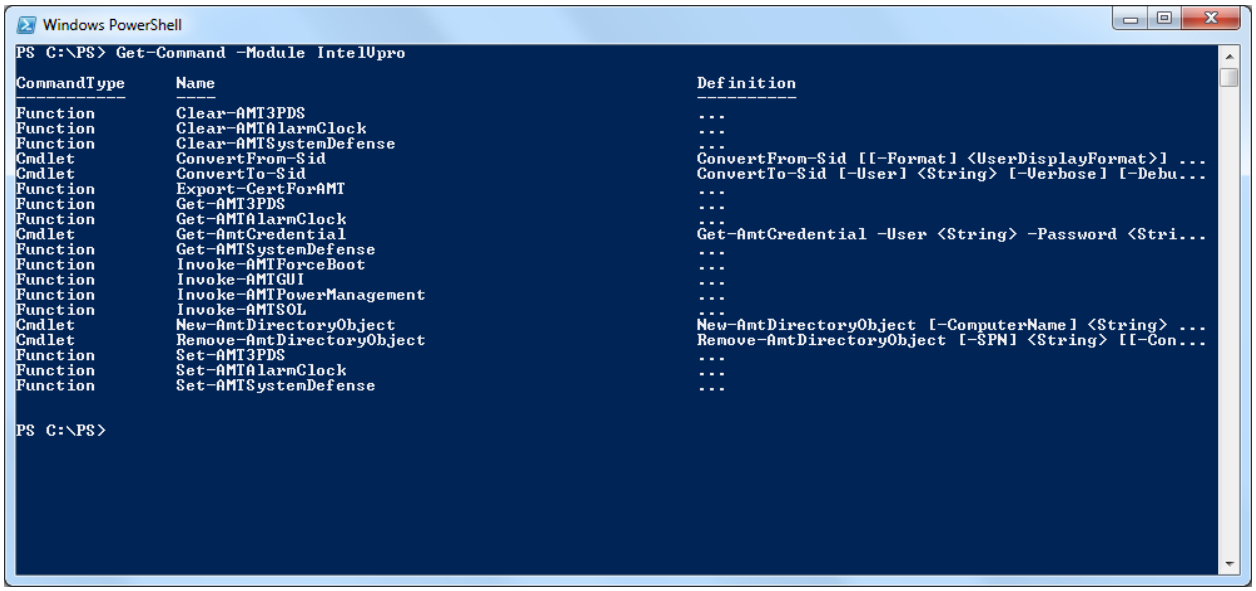


Figure 2: Listing the Available Cmdlets and Functions

4.2 Checking the Module Version

To view the manifest for the module in order to see the installed version, type **Get-Module -ListAvailable -name intelvpro | format-list**.

4.3 Usages

- To remotely power up, power down or power cycle the client Intel vPro system, use **Invoke-AMTPowerManagement** (see section 6.1.1).
- Need to perform remediation on a system? Use **Invoke-AMTForceBoot** (see section 6.2.1) to redirect the system's boot process, forcing it to boot from a network share, bootable CD-ROM or DVD, remediation drive, PXE or other boot device.
- To redirect the system's I/O via console redirection through serial-over-LAN (SOL), use **Invoke-AMTForceBoot** (section 6.2.1). This feature supports remote troubleshooting, remote repair, software upgrades, and similar processes.
- To access and change BIOS settings remotely, use **Invoke-AMTForceBoot** (section 6.2.1). Even if the system's power is off, the OS is down, or hardware has failed, you can still perform remote updates and corrections of configuration settings.

- To display persistent logs stored in protected memory, use **Get-AMTAccessMonitor** (see section 6.11.1) and **Get-AMTEventLog** (see section 6.11.2). The event log is available even if the OS is down or the hardware has already failed.

    - Store information in the Third Party data Store with **Set-AMT3PDS** (see section 6.6.1). For example, an Anti-Virus program could store version information in the protected memory that is available for out of band access. A different cmdlet could then use **Get-AMT3PDS** (see section 6.6.2) to identify systems that need updating. **NOTE:** The Intel AMT Third Party Data Storage (3PDS) feature was deprecated in Intel ME 11.0 firmware and removed in Intel ME 12.0.47.1524.

- To see what hardware a system has, use **Get-AMTHardwareAsset** (see section 6.11.4) to perform a hardware inventory. Hardware asset information is updated every time the system runs through power-on self-test (POST).

- The Intel vPro PowerShell Module does not natively support KVM Remote Control, but PowerShell can be used to start an application that does. For instance, to start RealVNC VNC* Viewer Plus, type the following:
  .\vncviewerplus.exe $ComputerName -amtusername=admin

5 Adding a New Script

This section describes how you can add a new PowerShell script to the Windows PowerShell Module for Intel vPro Technology.

To add a new script:

1. Create a PowerShell script file. Use the **invoke-flow.ps1** script template located in the PS_Snippets\Common\ directory and include a function that has the name of the script. For example, if you name the script file **Invoke-NewCommand.ps1**, the function should appear as follows:

   ```
   function Invoke-NewCommand
     {
          script body
          ....
     }
   ```

   Note that the **Import-Module IntelvPro** and **Remove-Module IntelvPro** commands in the **invoke-flow.ps1** template file are not needed and can be deleted from your script.

2. Place your new PowerShell script in the C:\Program Files (x86)\WindowsPowerShell\Modules\IntelvPro\ directory.
3. Edit the \Bin\IntelvPro**IntelvPro.psm1** file: Add the following line to enable the module to recognize the new script file:

   ```
   .$psScriptRoot\Invoke-NewCommand.ps1
   ```

   (replacing "Invoke-NewCommand" with the name of your script).

4. Open a PowerShellx86 window and run the following commands:
   a. **Import-Module IntelvPro**
   b. **Remove-Module IntelvPro**
   c. **Import-Module IntelvPro**

The module should now recognize the new command.

6 Cmdlet Information

Table 1: Cmdlet Support of Intel ME/CSME Versions

Version	Intel ME 3.0	Intel ME 3.2	Intel ME 5.1	Intel ME 6.0 through Intel ME 14.*	Intel CSME 15.0 and later
Invoke-AMTPowerManagement	X	X	X	X	X
Invoke-AMTForceBoot	X	X	X	X	X
Invoke-AMTSOL	X	X	X	X	X
Set-AMTAlarmClock			X	X	X
Get-AMTAlarmClock			X	X	X
Set-AMTSystemDefense	X	X	X	X	X
Clear-AMTSystemDefense	X	X	X	X	X
Set-AMT3PDS	X	X	X	X[1]	
Get-AMT3PDS	X	X	X	X[1]	
Clear-AMT3PDS	X	X	X	X[1]	
Invoke-AMTGUI	X	X	X	X	X
Get-AMTIDER	X	X	X	X	X
Start-AMTIDER	X	X	X	X	X
Stop-AMTIDER	X	X	X	X	X
Get-AMTAccessMonitor		X	X	X	X
Get-AMTEventLog		X	X	X	X
Get-AMTFirmwareVersion		X	X	X	X
Get-AMTHardwareAsset		X	X	X	X
Get-AMTPowerState		X	X	X	X
Get-UniquePlatformIDFeatureInfo					X

[1] No longer supported starting Intel CSME 12.0.47.1524

6.1 Intel AMT Power Management

Intel AMT Power Management can remotely power up, power down, or reset a client, independent of Operating System or hardware state.

6.1.1 Invoke-AMTPowerManagement

NAME

Invoke-AMTPowerManagement

SYNOPSIS

Invokes an Intel Active Management Technology power control command

SYNTAX

Invoke-AMTPowerManagement [-ComputerName] <String[]> [-Port] <String> [-Operation] <String>[-TLS] [-AcceptSelfSignedCert] [-Username <String>] [-Password <String>] [-Credential] <PSCredential>] [<CommonParameters>]

The valid parameters for **–Operation** are {PowerOn, PowerOff, Reset, Sleep, Hibernate, GracefulOff, GracefulReset, NMI, WakeFromConnectedStandby}.

DESCRIPTION

This cmdlet invokes an Intel Active Management Technology power control operation (Power On, Power Off, or Power Reset, Sleep, Hibernate, Graceful Off, Graceful Reset, NMI, or Wake from connected standby) from clients that have Intel AMT firmware version 3.0 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Invoke-AMTPowerManagement –Full

```
    -------------------------- EXAMPLE 1 --------------------------

    C:\PS>Invoke-AMTPowerManagement -computer:vproclient.vprodemo.com -TLS
-Operation:PowerOn


    ComputerName                 Port        Operation           Status
    ------------                 ----        ---------           ------
    vproclient.vprodemo.com      16993       PowerOn             Successful


    -------------------------- EXAMPLE 2 --------------------------

    C:\PS>Invoke-AMTPowerManagement vproclient -Operation:Reset -Username:amtuser

    Will prompt for digest username password.

    ComputerName                 Port        Operation           Status
```

```
    ------------                    ----        ---------           ------
    vproclient                      16992       Reset               Successful
    -------------------------- EXAMPLE 3 --------------------------

    C:\PS>Invoke-AMTPowerManagement vproclient.vprodemo.com -Operation PowerOff
-Username:vprodemo\ITHelpDesk


    Will prompt for Kerberos username password.

    ComputerName                    Port        Operation           Status
    ------------                    ----        ---------           ------
    vproclient.vprodemo.com         16992       PowerOff            Successful




    -------------------------- EXAMPLE 4 --------------------------

    C:\PS>Invoke-AMTPowerManagement -ComputerName:vproclient.vprodemo.com
-Operation:PowerOff -credential $AMTCredential -TLS

    ComputerName                    Port         Operation          Status
    ------------                    ----         ---------          ------
    vproclient.vprodemo.com         16993        PowerOff           Successful
```

6.2 Intel AMT Force Boot

The Intel AMT Force Boot cmdlet remotely boots a client to a specified boot device such as PXE, CD/DVD, or local hard drive.

6.2.1 Invoke-AMTForceBoot

NAME

Invoke-AMTForceBoot

SYNOPSIS

Invokes the Intel Active Management Technology force boot command

SYNTAX

Invoke-AMTForceBoot [-ComputerName] <String[]> [-Port] <String>] [-TLS] [-AcceptSelfSignedCert] [-Operation] <String> [-Device] <String> [[-IDERPath] <String>] [-Console] <String>] [[-SOLTerminalPath] <String>] [[-SOLTerminalArgList] <String>] [-Userna me <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

The valid parameters for **–Operation** are {PowerOn, Reset}.

The valid parameters for **–Device** are {HardDrive, PXE, Optical, IDER, BIOSSetup}.

- **HardDrive** forces a boot to the hard drive, regardless of inserted bootable media.
- **PXE** forces a PXE boot.
- **Optical** forces a boot from the media in the optical drive.
- **IDER** forces an Intel vPro IDE Redirection boot.
- **BIOSSetup** forces a boot to the BIOS Setup configuration screens.

The valid parameter for **–Console** is {SOL}.

If SOL is specified, a Serial Over LAN connection is made to the target system. The local endpoint of the serial session is 127.0.0.1. A path to a terminal program must be specified, along with arguments to invoke that terminal program. The **Invoke-AMTForceBoot** cmdlet will determine the port to connect to, so the argument list must define a '%Port' variable so that the cmdlet knows where to put the actual port number.

For example, to use Microsoft telnet, the following would be defined:

$SOLTerminalPath = "telnet"

$SOLTerminalArgList = "-t ANSI 127.0.0.1 %Port"

DESCRIPTION

This cmdlet invokes an Intel Active Management Technology forced boot to a PXE server, the local hard drive, CD/DVD ROM drive, or remote DVD/CD ISO image from clients that have Intel AMT firmware version 3.0 or higher.

For more details, review the PowerShell integrated help by typing:

Get-Help Invoke-AMTForceBoot –Full

```
    -------------------------- EXAMPLE 1 --------------------------

    C:\PS>Invoke-AMTForceBoot -Computer:vproclient.vprodemo.com -TLS
-Operation:PowerOn -Device:PXE


    ComputerName            Port   Operation       Device              Status
    ------------            ----   ---------       ------              ------
    vproclient.vprodemo.com 16993  PowerOn         PXE                 Successful




    -------------------------- EXAMPLE 2 --------------------------

    C:\PS>Invoke-AMTForceBoot 192.168.1.100 PowerOn PXE -credential $AMTCredential
-TLS


    ComputerName            Port   Operation       Device              Status
    ------------            ----   ---------       ------              ------
    192.168.1.100           16992  PowerOn         PXE                 Successful
```

```
    -------------------------- EXAMPLE 3 --------------------------

    C:\PS>Invoke-AMTForceBoot vproclient Reset -Device:Optical -Username:amtuser

    Will prompt for digest username password.

    ComputerName            Port   Operation        Device              Status
    ------------            ----   ---------        ------              ------
    vproclient              16992  Reset            Optical             Successful



    -------------------------- EXAMPLE 4 --------------------------

    C:\PS>Invoke-AMTForceBoot vproclient.vprodemo.com -Operation PowerOn
-Device:HardDrive -Username:vprodemo\ITHelpDesk

    Will prompt for Kerberos username password.

    ComputerName            Port   Operation        Device              Status
    ------------            ----   ---------        ------              ------
    vproclient.vprodemo.com 16992  PowerOn          HardDrive           Successful



    -------------------------- EXAMPLE 5 --------------------------

    C:\PS>Invoke-AMTForceBoot -Computer:vproclient.vprodemo.com -Operation:Reset
-Device:IDER -IDERPath:"C:\bootable_image.iso"

    ComputerName            Port   Operation        Device              Status
    ------------            ----   ---------        ------              ------
    vproclient.vprodemo.com 16992  PowerOn          IDER                Successful



    -------------------------- EXAMPLE 6 --------------------------

    C:\PS>Invoke-AMTForceBoot -ComputerName computer1.vprodemo.com,
doesnotexist.vprodemo.com -TLS -Operation Reset -Device:Optical | Where {$_.Status
-eq "Failed"}

    Will perform the power operation on every AMT client in the list, but only
display the ones that failed.

    ComputerName            Port   Operation        Device              Status
    ------------            ----   ---------        ------              ------
    doesnotexist.vprodem... 16993  Reset            Optical             Failed



   -------------------------- EXAMPLE 7 --------------------------

   C:\PS>Get-SomeDataFromOtherCMDLet | Select ComputerName | Invoke-AMTForceBoot
-TLS -Operation PowerOn -Device:HardDrive

   Get-SomeDataFromOtherCMDLet is a custom cmdlet that has an output of
ComputerName, Port, and Operation; however, you only select ComputerName.
Remaining parameters are manually provided.
```

```
    ComputerName            Port   Operation        Device              Status
    ------------            ----   ---------        ------              ------
computer1.vprodemo.com  16993                  PowerOn                 HardDrive
Successful
   computer2.vprodemo.com  16993                  PowerOn
HardDrive               Successful
   computer3.vprodemo.com  16993                  PowerOn
HardDrive               Successful



   ------------------------- EXAMPLE 8 -------------------------

   C:\PS>Invoke-AMTForceBoot vproclient.vprodemo.com -Operation:Reset
-Device:BIOSSetup -Credential $AMTCredential -Console SOL -SOLTerminalPath "telnet"
-SOLTerminalArgList "-t ANSI 127.0.0.1 %Port"


   This will reboot the client to the BIOS Setup screens while connecting SOL to a
telnet window.

   Ok

   ComputerName : 192.168.1.106
   Port         : 16992
   Operation    : reset
   Device       : BIOSSetup
   Status       : Successful
```

6.3 Intel AMT Serial Over LAN

Serial Over LAN (SOL) is an Intel AMT capability that enables the input and output of the serial port of a managed system to be sent over the network. Console redirection can be performed over this SOL interface.

6.3.1 Invoke-AMTSOL

NAME

Invoke-AMTSOL

SYNOPSIS

Establishes a Serial Over LAN (SOL) session

SYNTAX

Invoke-AMTSOL [-ComputerName] <String[]> [-Port] <String> [-TLS] [-AcceptSelfSignedCert] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet establishes a Serial Over LAN communication to interact with clients that have Intel AMT firmware version 3.0 or higher.

For more details, review the PowerShell integrated help by typing:

Get-Help Invoke-AMTSOL –Full

```
------------------------- EXAMPLE 1 -------------------------

C:\PS>Invoke-AMTSOL -computer:vproclient.vprodemo.com -username:admin

------------------------- EXAMPLE 2 -------------------------

C:\PS>Invoke-AMTSOL 192.168.1.100 -credential $AMTCredential –TLS
```

To terminate a SOL session:

Press the ESC key.

6.4 Intel AMT Alarm Clock

The Intel AMT Alarm Clock can be configured to wake a managed system once at a specific time or multiple times with a periodical interval.

6.4.1 Set-AMTAlarmClock

NAME

Set-AMTAlarmClock

SYNOPSIS

Sets an Intel® Active Management Technology alarm clock timer.

SYNTAX

Set-AMTAlarmClock [-ComputerName] <String[]> [-Port <String>] [-AlarmTime] <String> [-Interval <String>] [-AlarmName <String>] [-DeleteCompletion] [-TLS] [-AcceptSelfSignedCert] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

AlarmTime:

The AlarmTime parameter is the date and time to wake the client. It is set as [YYYY]-[MM]-[DD]T[HH]:[MM]:[SS] Example: 2010-07-14T02:00:00 would wake the client on July 14, 2010 @ 02:00.

If the user sets the alarm date and time on a client in a different time zone, the alarm time specified will be set to the proper GMT for the client. For example, if running the

cmdlet from a client in the Pacific Time zone to configure the wake up time on a client configured with Eastern Time for 08:00, the alarm clock will wake the client at 08:00 Eastern time.

Interval:

Interval parameter is the desired reoccurrence interval for the alarm to be set. The format is: [DD]-[HH]-[MM]-[SS] Example: 07-00:00:00 would have a reoccurrence of every seven day at the same time. Example: 00-02:30:00 would have a reoccurrence of every 2 hours 30 minutes.

DESCRIPTION

This CmdLet allows the user to set a wake timer on clients that have Intel Active Management Technology (AMT) firmware version 5.1 or higher.

Since Intel AMT firmware version 8.0 or higher, multiple alarm clock timers are supported, so when setting an alarm clock, an alarm name must be provided.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Set-AMTAlarmClock -Full

```
    -------------------------- EXAMPLE 1 --------------------------

    C:\PS>Set-AMTAlarmClock -ComputerName:vproclient.vprodemo.com -TLS
-AlarmTime:2010-07-14T02:00:00


    Sets one time occurrence for wake up alarmclock.

    ComputerName  : vproclient.vprodemo.com
    Port          : 16993
    NextAlarmTime : Wednesday, July 14, 2010 2:00:00 AM
    Status        : Successful

     -------------------------- EXAMPLE 2 --------------------------

     C:\PS>Set-AMTAlarmClock -ComputerName:vproclient.vprodemo.com -TLS
-AlarmTime:2010-07-14T02:00:00 -AlarmName MyDefaultAlarm -DeleteCompletion


     Sets one time occurrence for wake up alarmclock, the alarm name will be
MyDefaultAlarm and it will be deleted automaticly once it occurrs.

 ComputerName  : vproclient.vprodemo.com
 Port          : 16993
 NextAlarmTime : Wednesday, July 14, 2010 2:00:00 AM


    -------------------------- EXAMPLE 3 --------------------------

    C:\PS>Set-AMTAlarmClock vproclient.vprodemo.com -TLS -AlarmTime:2010-07-
14T02:00:00 -UserName vprodemo\administrator


    Will prompt for Kerberos User Password then Sets one time occurrence for wake
up alarmclock.

    ComputerName  : vproclient.vprodemo.com
    Port          : 16993
```

```
    NextAlarmTime : Wednesday, July 14, 2010 2:00:00 AM
    Status        : Successful



 -------------------------- EXAMPLE 4 --------------------------

    C:\PS>Set-AMTAlarmClock vproclient -UserName:admin -AlarmTime:2010-07-
14T02:00:00 -Interval:07-00:00:00


    Will prompt for Digest User Password then sets reoccuring wake up alarmclock
for once a week at that time.

    ComputerName     : vproclient
    Port             : 16992
    NextAlarmTime    : Wednesday, July 14, 2010 2:00:00 AM
    PeriodicInterval : P0Y0M07DT00H00M
    Status           : Successful



    -------------------------- EXAMPLE 5 --------------------------

    C:\PS>Get-Content computers.txt | Set-AMTAlarmClock -credential $AMTCredential
-TLS -AlarmTime:2010-07-14T02:00:00 -Interval:00-01:00:00 -credential
$SomeStoredPSCredential


    Will pull the list of amt clients from a text file and pipe them in the Set-
AMTAlarmClock CMDLet.
    Sets reoccuring wake up alarmclock for once every hour on and after that time.

    ComputerName     : computer1.vprodemo.com
    Port             : 16993
    NextAlarmTime    : Wednesday, July 14, 2010 2:00:00 AM
    PeriodicInterval : P0Y0M00DT02H00M
    Status           : Successful

    ComputerName     : computer2.vprodemo.com
    Port             : 16993
    NextAlarmTime    : Wednesday, July 14, 2010 2:00:00 AM
    PeriodicInterval : P0Y0M00DT02H00M
    Status           : Successful
```

6.4.2 Get-AMTAlarmClock

NAME

Get-AMTAlarmClock

SYNOPSIS

Returns status of the Intel Active Management Technology alarm clock timers

SYNTAX

Get-AMTAlarmClock [-ComputerName] <String[]> [-Port] <String> [-TLS] [-AcceptSelfSignedCert] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet returns the status of Intel Active Management Technology alarm clock timers from clients that have Intel AMT firmware version 5.1 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTAlarmClock -Full

```
    ------------------------- EXAMPLE 1 -------------------------

    C:\PS>Get-AMTAlarmClock -computer:vproclient.vprodemo.com -TLS


    ComputerName     : vproclient.vprodemo.com
    Port             : 16993
    NextAlarmTime    : Wednesday, July 14, 2010 2:00:00 AM
    PeriodicInterval : [None Set]
    Status           : Successful



    ------------------------- EXAMPLE 2 -------------------------

    C:\PS>Get-AMTAlarmClock vproclient -Username:amtuser -TLS


    Will prompt for digest username password.

    ComputerName     : vproclient
    Port             : 16993
    NextAlarmTime    : Wednesday, July 14, 2010 2:00:00 AM
    PeriodicInterval : 7 days, 0 hours, 0 minutes, 0 seconds
    Status           : Successful



    ------------------------- EXAMPLE 3 -------------------------

    C:\PS>Get-AMTAlarmClock vproclient.vprodemo.com -Username
vprodemo\administrator-TLS


    Will prompt for Kerberos username password.

    ComputerName     : vproclient
    Port             : 16993
    NextAlarmTime    : Wednesday, July 14, 2010 2:00:00 AM
    PeriodicInterval : [None Set]
    Status           : Successful



    ------------------------- EXAMPLE 4 -------------------------

    C:\PS>Get-AMTAlarmClock -ComputerName:vproclient.vprodemo.com -credential
$AMTCredential -TLS
```

```
    ComputerName     : vproclient
    Port             : 16993
    NextAlarmTime    : Wednesday, July 14, 2010 2:00:00 AM
    PeriodicInterval : [None Set]
    Status           : Successful



    -------------------------- EXAMPLE 5 -------------------------

    C:\PS>Get-AMTAlarmClock -ComputerName
computer1.vprodemo.com,doesnotexist.vprodemo.com -TLS | Where {$_.Status -eq
    "Failed"}


    Will perform the clear Alarm clock operation on every AMT client in the list,
but only display the ones that failed
    .

    ComputerName     : doesnotexist.vprodemo.com
    Port             : 16993
    Status           : Failed
    NextAlarmTime    : [None Set]
    PeriodicInterval : [None Set]



    -------------------------- EXAMPLE 6 -------------------------

    C:\PS>Get-Content computers.txt | Get-AMTAlarmClock -Port:16993


    Will pull the list of amt clients from a text file and pipe them into Get-
AMTAlarmClock.

    ComputerName     : computer1.vprodemo.com
    Port             : 16993
    NextAlarmTime    : [None Set]
    PeriodicInterval : [None Set]
    Status           : Successful

    ComputerName     : computer2.vprodemo.com
    Port             : 16993
    NextAlarmTime    : Wednesday, July 14, 2010 2:00:00 AM
    PeriodicInterval : 7 days, 0 hours, 0 minutes, 0 seconds
    Status           : Successful

    ComputerName     : computer3.vprodemo.com
    Port             : 16993
    NextAlarmTime    : Wednesday, July 14, 2010 2:00:00 AM
    PeriodicInterval : [None Set]
    Status           : Successful
```

6.4.3 Clear-AMTAlarmClock

NAME

Clear-AMTAlarmClock

SYNOPSIS

Clears Intel Active Management Technology alarm clock timers

SYNTAX

Clear-AMTAlarmClock [-ComputerName] <String[]> [-Port <String>] [-TLS] [-AcceptSelfSignedCert] [-AlarmName <String>] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet clears the Intel Active Management Technology alarm clock timers from clients that have Intel AMT firmware version 5.1 or higher.

From AMT firmware version 8.0 or higher, user can supply a specific alarm name to delete. Otherwise, all alarm timers will be deleted.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Clear-AMTAlarmClock -Full

```
C:\PS>Clear-AMTAlarmClock -computer:vproclient.vprodemo.com

ComputerName     : vproclient.vprodemo.com
Port             : 16992
NextAlarmTime    : [None Set]
PeriodicInterval : [None Set]
Status           : Successful

 ------------------------- EXAMPLE 2 -------------------------

 C:\PS>Clear-AMTAlarmClock -computer:vproclient.vprodemo.com -AlarmName
MyDefaultAlarm

 Will delete only the alarm named: MyDefaultAlarm

 ComputerName     : vproclient.vprodemo.com
 Port             : 16992
 AlarmName        : MyDefaultAlarm
 NextAlarmTime    : [None Set]
 PeriodicInterval : [None Set]
 DeleteOption     : [None Set]
 Status           : Successful

 ComputerName     : vproclient.vprodemo.com
 Port             : 16992
 NextAlarmTime    : [None Set]
 PeriodicInterval : [None Set]
 Status           : Successful

------------------------- EXAMPLE 3 -------------------------

C:\PS>Clear-AMTAlarmClock vproclient -Username:amtuser

Will prompt for digest username password.

ComputerName     : vproclient
Port             : 16992
NextAlarmTime    : [None Set]
PeriodicInterval : [None Set]
```

```
    Status           : Successful



    -------------------------- EXAMPLE 4 --------------------------

    C:\PS>Clear-AMTAlarmClock vproclient.vprodemo.com -Username
vprodemo\administrator


    Will prompt for Kerberos username password.

    ComputerName     : vproclient.vprodemo.com
    Port             : 16993
    NextAlarmTime    : [None Set]
    PeriodicInterval : [None Set]
    Status           : Successful



    -------------------------- EXAMPLE 5 --------------------------

    C:\PS>Clear-AMTAlarmClock -ComputerName:vproclient.vprodemo.com -credential
$AMTCredential -TLS


    ComputerName     : vproclient.vprodemo.com
    Port             : 16993
    NextAlarmTime    : [None Set]
    PeriodicInterval : [None Set]
    Status           : Successful



    -------------------------- EXAMPLE 6 --------------------------

    C:\PS>Clear-AMTAlarmClock -ComputerName
computer1.vprodemo.com,doesnotexist.vprodemo.com | Where {$_.Status -eq "Fa
    iled"}


    Will perform the clear Alarm clock operation on every AMT client in the list,
but only display the ones that failed.

    ComputerName     : doesnotexist.vprodemo.com
    Port             : 16992
    Status           : Failed
    NextAlarmTime    : [None Set]
    PeriodicInterval : [None Set]



    -------------------------- EXAMPLE 7 --------------------------

    C:\PS>Get-Content computers.txt | Clear-AMTAlarmClock -TLS


    Will pull the list of amt clients from a text file and pipe them into the
Clear-AMTAlarmClock CMDLet.

    ComputerName     : computer1.vprodemo.com
    Port             : 16993
    NextAlarmTime    : [None Set]
    PeriodicInterval : [None Set]
    Status           : Successful

    ComputerName     : computer2.vprodemo.com
    Port             : 16993
    NextAlarmTime    : [None Set]
    PeriodicInterval : [None Set]
```

```
Status           : Successful

ComputerName     : computer3.vprodemo.com
Port             : 16993
NextAlarmTime    : [None Set]
PeriodicInterval : [None Set]
Status           : Successful
```

6.5 Intel AMT System Defense

System Defense is an Intel AMT capability that enforces network security policies such as filtering and preventing network traffic from reaching the operating system, while still managing the client Out of Band with Intel AMT.

6.5.1 Set-AMTSystemDefense

NAME

Set-AMTSystemDefense

SYNOPSIS

Enables Intel Active Management Technology System Defense

SYNTAX

Set-AMTSystemDefense [-ComputerName] <String[]> [-Port] <String> [-TLS] [-AcceptSelfSignedCert] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet configures network access using Intel Active Management Technology System Defense from clients that have Intel AMT Firmware version 3.0 and Higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Set-AMTSystemDefense –Full

```
-------------------------- EXAMPLE 1 --------------------------

C:\PS>Set-AMTSystemDefense vproclient.vprodemo.com -TLS

ComputerName                Port        Status       EnabledOnInterfaces
------------                ----        ------       -------------------
vproclient.vprodemo.com     16993       Successful   Wireless1, Wired
```

```
    -------------------------- EXAMPLE 2 --------------------------

    C:\PS>Set-AMTSystemDefense vproclient 16992  -username:amtuser

    Will prompt for digest username password.

    ComputerName                 Port        Status       EnabledOnInterfaces
    ------------                 ----        ------       -------------------
    vproclient                   16992       Successful   Wireless1, Wired0



    -------------------------- EXAMPLE 3 --------------------------

    C:\PS>Get-Content computers.txt | Set-AMTSystemDefense -TLS

    Will pull the list of amt clients from a text file and pipe them in the set-
AMTSystemDefense Cmdlet.

    ComputerName                 Port        Status       EnabledOnInterfaces
    ------------                 ----        ------       -------------------
    Computer1.vprodemo.com       16993       Successful   Wireless1, Wired0
    Computer2.vprodemo.com       16993       Successful   Wired0

    -------------------------- EXAMPLE 4 --------------------------

    C:\PS>Set-AMTSystemDefense vproclient 16992 –XMLConfig xmlfile.xml
```

An XMLConfig switch may be passed in to invoke user defined network traffic filters with an xml file. For more details on xml file format, refer to section 6.5.4.

NOTE

If no configure xml file is specified, the default behavior is to block all incoming and outgoing network traffic.

6.5.2 Get-AMTSystemDefense

NAME

Get-AMTSystemDefense

SYNOPSIS

Returns status of Intel Active Management Technology System Defense policies

SYNTAX

Get-AMTSystemDefense [-ComputerName] <String[]> [-Port] <String> [-TLS] [-AcceptSelfSignedCert] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet returns the status of Intel Active Management Technology System Defense network access policies from clients that have Intel AMT firmware version 3.0 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTSystemDefense –Full

```
    -------------------------- EXAMPLE 1 --------------------------

   C:\PS>Get-AMTSystemDefense vproclient.vprodemo.com -TLS

    ComputerName                 Port        Status       EnabledOnInterfaces
    ------------                 ----        ------       -------------------
    vproclient.vprodemo.com      16993       Successful   [None]



   -------------------------- EXAMPLE 2 --------------------------

   C:\PS>Get-AMTSystemDefense vproclient 16992  -username:amtuser

   Will prompt for digest username password.
    ComputerName                 Port        Status       EnabledOnInterfaces
    ------------                 ----        ------       -------------------
   vproclient                   16992       Successful   Wireless1, Wired0



   -------------------------- EXAMPLE 3 --------------------------

   C:\PS>Get-AMTSystemDefense -ComputerName:vproclient.vprodemo.com -credential
$AMTCredential -TLS
    ComputerName                 Port        Status       EnabledOnInterfaces
    ------------                 ----        ------       -------------------
    vproclient.vprodemo.com      16993       Successful   Wired0



   -------------------------- EXAMPLE 4 --------------------------

   C:\PS>Get-Content computers.txt | Set-AMTSystemDefense -TLS

   Will pull the list of amt clients from a text file and pipe them into set-
AMTSystemDefense.
```

```
ComputerName                Port        Status       EnabledOnInterfaces
------------                ----        ------       -------------------
Computer1.vprodemo.com      16993       Successful   Wireless1, Wired0
Computer2.vprodemo.com      16993       Successful   [None]
```

6.5.3 Clear-AMTSystemDefense

NAME

Clear-AMTSystemDefense

SYNOPSIS

Clears the Intel Active Management Technology System Defense policy

SYNTAX

Clear-AMTSystemDefense [-ComputerName] <String[]> [-Port <String>] [-TLS] [-AcceptSelfSignedCert] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet clears the Intel Active Management Technology network access policy from clients that have Intel AMT firmware version 3.0 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Clear-AMTSystemDefense –Full

```
-------------------------- EXAMPLE 1 --------------------------

C:\PS>Clear-AMTSystemDefense vproclient.vprodemo.com -TLS

ComputerName                Port        Status       EnabledOnInterfaces
------------                ----        ------       -------------------
vproclient.vprodemo.com     16993       Successful   [None]



-------------------------- EXAMPLE 2 --------------------------

C:\PS>Clear-AMTSystemDefense vproclient 16992 -username:amtuser

Will prompt for digest username password.
ComputerName                Port        Status       EnabledOnInterfaces
```

```
    ------------                    ----        ------      -------------------
    vproclient                      16992       Successful  [None]



    -------------------------- EXAMPLE 3 -------------------------

    C:\PS>Clear-AMTSystemDefense -ComputerName:vproclient.vprodemo.com -credential
$AMTCredential -TLS

    ComputerName                    Port        Status      EnabledOnInterfaces
    ------------                    ----        ------      -------------------
    vproclient.vprodemo.com         16993       Successful  [None]



    -------------------------- EXAMPLE 4 -------------------------

    C:\PS>Get-Content computers.txt | Clear-AMTSystemDefense -TLS

    Will pull the list of clients from a text file and pipe them into Clear-
AMTSystemDefense.

    ComputerName                    Port        Status      EnabledOnInterfaces
    ------------                    ----        ------      -------------------
    vproclient1.vprodemo.com        16993       Successful  [None]
    vproclient2.vprodemo.com        16993       Successful  [None]
```

6.5.4 XML Format

A System Defense policy contains a set of filters that are applied to incoming and outgoing network packets, combined with actions to take when a packet matches or does not match the conditions in the filter.

Policy Supported Fields	Meaning
InstanceID	Enter any value (the value in this field is overridden by the program)
PolicyName	"ExamplePolicy" - Enter a meaningful name that you can use later to search for this instance. Maximum length 16.
PolicyPrecedence	In case multiple policies are being activated simultaneously, the policy with the highest precedence value takes effect
AntiSpoofingSupport	Anti Spoofing has the highest priority for blocking
FilterCreationHandles	A list of Filter Creation Handles to be included in the Policy

Policy Supported Fields	Meaning
TxDefaultDrop	Specifies whether the TX packet should be dropped on filter match
TxDefaultMatchEvent	Specifies whether an Event should be created in the Event Manager when this filter is matched
Tx DefaultCount	Specifies whether to count filter matches
RxDefaultDrop	Specifies whether the RX packet should be dropped on filter match
RxDefaultMatchEvent	Specifies whether an Event should be created in the Event Manager when this filter is matched
RxDefaultCount	Specifies whether to count filter matches

You can create two types of System Defense filters:

- Ethernet Filter
- IP Filter

Ethernet Filter belongs to the class AMT_Hdr8021Filter. The 8021Filter allows 802.1 source and destination MAC addresses, as well as the 802.1 protocol ID, priority, and VLAN identifier fields, to be expressed in a single object to classify and identify traffic.

AMT_Hdr8021Filter Supported Fields	Meaning
Name	Defines the label by which the Filter Entry is known and uniquely identified
PolicyName	The name of the policy that this filter will be used in.
CreationClassName	Indicates the name of the class or the subclass used in the creation of an instance
SystemName	The scoping ComputerSystem's Name
SystemCreationClassName	The scoping ComputerSystem's CreationClassName
HdrProtocolID8021	This property is a 16-bit unsigned integer, representing an Ethernet protocol type
FilterProfile	Specifies the type of behavior exhibited by the filter
FilterDirection	Specifies the traffic direction (transmit or receive) that the filter governs

AMT_Hdr8021Filter Supported Fields	Meaning
ActionEventOnMatch	Specifies whether an Event should be created in the Event Manager when this filter is matched
FilterProfileData	An extra data parameter which is used depending on the FilterProfile: It is left blank for Drop/Pass/Statistics filters, but is required for Rate Limit filters

IPFilter belongs to the class AMT_IPHeadersFilter. This filter contains the most commonly required properties for performing filtering on IP, TCP or UDP headers. Properties in an instance of the IPHeadersFilter are treated as 'all values'.

AMT_IPHeadersFilter Supported Fields	Meaning
Name	Defines the label by which the Filter Entry is known and uniquely identified
PolicyName	The name of the policy in which this filter will be used.
CreationClassName	Indicates the name of the class or the subclass used in the creation of an instance
SystemName	The scoping ComputerSystem's Name
SystemCreationClassName	The scoping ComputerSystem's CreationClassName
HdrIPVersion	Identifies the version of the IP addresses for IP header filters
HdrSrcAddress	An OctetString, of a size determined by the value of the HdrIPVersion property, representing a source IP address

AMT_IPHeadersFilter Supported Fields	Meaning
HdrSrcMask	An OctetString, of a size determined by the value of the HdrIPVersion property, representing a mask to be used in comparing the source address in the IP header with the value represented by the HdrSrcAddress property
HdrDestAddress	An OctetString, of a size determined by the value of the HdrIPVersion property, representing a destination IP address
HdrDestMask	An OctetString, of a size determined by the value of the HdrIPVersion property, representing a mask to be used in comparing the destination address in the IP header with the value represented in the HdrDestAddress property
HdrProtocolID	8-bit unsigned integer, representing an IP protocol type
HdrSrcPortStart	Represents the lower end of a range of UDP or TCP source ports
HdrSrcPortEnd	Represents the upper end of a range of UDP or TCP source ports
HdrDestPortStart	Represents the lower end of a range of UDP or TCP destination ports
HdrDestPortEnd	Represents the upper end of a range of UDP or TCP destination ports
TCPFlagsOn	A set of flags whose effective value in the TCP header of each packet must be ON for filter to take effect
TCPFlagsOff	A set of flags whose effective value in the TCP header of each packet must be OFF for filter to take effect
FilterProfile	Specifies the type of behavior exhibited by the filter
FilterDirection	Specifies the traffic direction (transmit or receive) that the filter governs
ActionEventOnMatch	Specifies whether an Event should be created in the Event Manager when this filter is matched
FilterProfileData	An extra data parameter which is used depending on the FilterProfile: It is left blank for Drop/Pass/Statistics filters, but is required for Rate Limit filters

6.5.5 Example: Policy Blocking All Traffic

Following is an example of a policy that blocks **all** incoming and outgoing network traffic:

```
<?xml version="1.0"?>
<SystemDefensePolicySet>
<ArrayOfFilter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<Filter xsi:type="AMT_Hdr8021Filter">
    <Name>defaultBlock</Name>
    <PolicyName>defaultPolicy</PolicyName>
    <filterSchema>http://intel.com/wbem/wscim/1/amt-
schema/1/AMT_Hdr8021Filter</filterSchema>
    <CreationClassName>n/a</CreationClassName>
    <SystemName>n/a</SystemName>
    <SystemCreationClassName>n/a</SystemCreationClassName>
    <FilterProfile>1</FilterProfile>
    <FilterDirection>0</FilterDirection>
    <ActionEventOnMatch>false</ActionEventOnMatch>
    <HdrProtocolID8021>2048</HdrProtocolID8021>
  </Filter>
</ArrayOfFilter>
<ArrayOfPolicies xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <Policy xsi:type="AMT_systemDefensePolicy">
        <PolicyName>defaultPolicy</PolicyName>
                <PolicyPrecedence>0</PolicyPrecedence>
                <AntiSpoofingSupport>3</AntiSpoofingSupport>
                <TxDefaultDrop>false</TxDefaultDrop>
                <TxDefaultMatchEvent>false</TxDefaultMatchEvent>
                <TxDefaultCount>true</TxDefaultCount>
                <RxDefaultDrop>false</RxDefaultDrop>
                <RxDefaultMatchEvent>true</RxDefaultMatchEvent>
                <RxDefaultCount>false</RxDefaultCount>
                <Active>true</Active>
        </Policy>
</ArrayOfPolicies>
</SystemDefensePolicySet>
```

For more details, refer to the System Defense section in the AMT SDK:http://software.intel.com/sites/manageability/AMT_Implementation_and_Reference_Guide/default.htm

6.6 Intel AMT Third Party Data Storage (3PDS)

The Intel AMT Third Party Data Storage (3PDS) is a persistent, nonvolatile memory space available to write and read data even when the OS is unresponsive or when management agents are missing.

NOTE

The Intel AMT Third Party Data Storage (3PDS) feature was deprecated in Intel ME 11.0 firmware and removed in Intel ME 12.0.47.1524.

6.6.1 Set-AMT3PDS

NAME

Set-AMT3PDS

SYNOPSIS

Stores data in the Intel Active Management Technology Third Party Data Storage (3PDS)

SYNTAX

Set-AMT3PDS [-ComputerName] <String[]> [-Port] <String> [-Operation] <String> [-Enterprise] <String> [-Vendor] <String> [-Application] <String> [-Block] <String> [[-BlockData] <String>] [-BlockHidden <Boolean>] [-AppendWrite <Boolean>] [-TLS] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

The valid parameters for **–Operation** are {Create, Write, CreateWrite}.

- **Create** makes a new block.
- **Write** writes the data to the block. All data is overwritten unless the **-AppendWrite** switch is specified.
- **CreateWrite** is a combination of the two operations: create a new block and write data to it.

Understanding 3PDS structure:

Data stored within the 3PDS is stored within blocks of nonvolatile memory in a hierarchical structure. Each block must be associated with a tiered structure of Enterprise -> Vendor -> Application -> Block Name.

3PDS Machine UUID:

When a block is created, the application that created the block will specify a GUID to identify itself as the entity that created the block. When modifying blocks that were created by a different entity it may be necessary to specify the Machine UUID as part of the cmdlet parameter.

DESCRIPTION

This cmdlet stores data in the Intel Active Management Technology Third Party Data Storage (3PDS) of clients that have Intel® AMT firmware version 3.0 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Set-AMT3PDS –Full

```
    -------------------------- EXAMPLE 1 --------------------------

    C:\PS>Set-AMT3PDS -computer:vproclient.vprodemo.com -TLS -Operation:Create
-Enterprise:Intel -Vendor:Intel -Application:PowerShell -Block:TestName


    Creates Block using Kerberos credentials.

    ComputerName                 Port             Operation          Status
    ------------                 ----             ---------          ------
    vproclient.vprodemo.com      16993            Create             Successful




    -------------------------- EXAMPLE 2 --------------------------

    C:\PS>Set-AMT3PDS 192.168.1.100 Write -credential $AMTCredential -
Enterprise:Intel -Vendor:Intel -Application:Power
    Shell -Block:TestName -BlockData:"This is test"


    Creates Block and write data to block

    ComputerName                 Port             Operation          Status
    ------------                 ----             ---------          ------
    192.168.1.100                16992            Create             Successful



    -------------------------- EXAMPLE 3 --------------------------

    C:\PS>Set-AMT3PDS -ComputerName:vproclient.vprodemo.com -TLS -Operation:Write
-Enterprise:Intel -Vendor:Intel -Application:PowerShell -Block:TestName -
BlockData:"Append this to existing data in block" -AppendWrite $true
```

```
    Appends the data to data in existing block.

    ComputerName                 Port             Operation          Status
    ------------                 ----             ---------          ------
    vproclient.vprodemo.com      16993            Write              Successful



    -------------------------- EXAMPLE 4 --------------------------

    C:\PS>Get-Content computers.txt | Set-AMT3PDS -TLS -Operation:Write
-Enterprise:Intel -Vendor:Intel -Application:PowerShell -Block:TestName
-BlockData:"This is test"


    Will pull the list of amt clients from a text file and pipe them in the Set-
AMT3PDS CMDLet.
```

6.6.2 Get-AMT3PDS

NAME

Get-AMT3PDS

SYNOPSIS

Retrieves data from the Intel Active Management Technology Third Party Data Storage

SYNTAX

Get-AMT3PDS [-ComputerName] <String[]> [-Port] <String> [-Operation] <String> [[-Enterprise] <String>] [[-Vendor] <String>] [[-Application] <String>] [[-Block] <String>] [[-MachineUUID] <String>] [-TLS] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

The valid parameters for **–Operation** are {Read, ListBlocks}.

- **Read** reads the data.
- **ListBlocks** retrieves all the available blocks.

DESCRIPTION

This cmdlet enables the user to retrieve data from Intel® Active Management Technology Third Party Data Storage (3PDS) from clients that have Intel® AMT firmware version 3.0 or higher.

When accessing the 3PDS, the Enterprise, Vendor, Application must be set. A UUID may be optional. When reading data from the 3PDS, the Get-AMT3PDS cmdlet will

have read access to all blocks made using the same Enterprise, Vendor, and Application.

Understanding 3PDS structure:

Data stored within the 3PDS is stored within blocks of nonvolatile memory in a hierarchical structure. Each block must be associated to a tiered structure of Enterprise -> Vendor -> Application -> Block Name.

3PDS Machine UUID:

When a block is created, the application that created the block specifies a GUID to identify itself as the entity that created the block. When modifying blocks that were created by a different entity, it may be necessary to specify the Machine UUID as part of the cmdlet parameter.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMT3PDS –Full

```
    -------------------------- EXAMPLE 1 -------------------------

    C:\PS>Get-AMT3PDS -computerName:vproclient.vprodemo.com -TLS
-Operation:ListBlocks


    Retrieves all the available blocks.

    ComputerName   : vproclient.vprodemo.com
    Port           : 16993
    Operation      : listblocks
    Status         : Success
    UUID           : BEAEE8BF2C09406DAE533F16080D2A6F
    Enterprise     : Virtualization
    Vendor         : Microsoft
    Application    : System Center ConfigMgr
    BlockName      : Out Of Band Management
    NumberOfBlocks : 1


    -------------------------- EXAMPLE 2 -------------------------

    C:\PS>Get-AMT3PDS 192.168.1.100 -Operation ListBlocks -credential
$AMTCredential -TLS


    ComputerName   : 192.168.1.100
    Port           : 16993
    Operation      : listblocks
    Status         : Success
    UUID           : A1DF77DC16E2469188B2E1F389E5A472
    Enterprise     : Intel
    Vendor         : Intel
    Application    : PowerShell
    BlockName      : Test
    NumberOfBlocks : 1


    -------------------------- EXAMPLE 3 -------------------------

    C:\PS>Get-AMT3PDS vproclient.vprodemo.com -Operation:Read
-Username:vprodemo\ITHelpDesk -Enterprise:Intel -Vendor:Intel
```

```
-Application:PowerShell -Block:Test


    Will prompt for Kerberos username password and then retrieve Data.

    ComputerName : vproclient.vprodemo.com
    Port         : 16992
    Operation    : read
    Status       : Success
    Blocks       : 1
    Data         : Test Data


    -------------------------- EXAMPLE 4 --------------------------

    C:\PS>Get-Content computers.txt | Get-AMT3PDS -TLS -Operation:ListBlocks


    Will pull the list of amt clients from a text file and pipe them in Get-
AMT3PDS.



    -------------------------- EXAMPLE 5 --------------------------

    C:\PS>Get-AMT3PDS-computerName:vproclient.vprodemo.com -port:16993
-Operation:read -Enterprise:"Virtualization" -Vendor:"Microsoft"
-Application:"System Center ConfigMgr"
-Block:"Out Of Band Management"
-MachineUUID:"BEAEE8BF2C09406DAE533F16080D2A6F"


    Example to pull data from the AMT 3PDS accessible by System Center
Configuration Manager
```

6.6.3 Clear-AMT3PDS

NAME

Clear-AMT3PDS

SYNOPSIS

Deletes data from the Intel® Active Management Technology Third Party Data Storage.

SYNTAX

Clear-AMT3PDS [-ComputerName] <String[]> [-Port] <String> [-Enterprise] <String> [[-Vendor] <String>] [[-Application] <String>] [[-Block] <String>] [[-MachineUUID] <String>] [-TLS] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet deletes data from the Intel® Active Management Technology Third Party Data Storage (3PDS) from clients that have Intel® AMT firmware version 3.0 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Clear-AMT3PDS –Full

```
    ------------------------- EXAMPLE 1 -------------------------
    C:\PS>Clear-AMT3PDS -computer:vproclient.vprodemo.com -TLS -Enterprise:Intel


    Will delete all block data under enterprise value specified.

    ComputerName                 Port            Operation          Status
    ------------                 ----            ---------          ------
    vproclient.vprodemo.com      16993           Delete             Successful


    ------------------------- EXAMPLE 2 -------------------------

    C:\PS>Clear-AMT3PDS 192.168.1.100 -credential $AMTCredential -Enterprise:Intel
-Vendor:Intel


    Will delete all block data under Enterprise, Vendor value specified.

    ComputerName                 Port            Operation          Status
    ------------                 ----            ---------          ------
    192.168.1.100                16992           Delete             Successful



    ------------------------- EXAMPLE 3 -------------------------

    C:\PS>Clear-AMT3PDS vproclient 16992 -Username:amtuser -Enterprise:Intel
-Vendor:Intel -Application:PowerShell


    Will delete all block data under Enterprise, Vendor, Application value
specified.

    ComputerName                 Port            Operation          Status
    ------------                 ----            ---------          ------
    vproclient                   16992           Delete             Successful


    ------------------------- EXAMPLE 4 -------------------------

    C:\PS>Clear-AMT3PDS vproclient.vprodemo.com -credential $AMTCredential
-Enterprise:Intel -Vendor:Intel -Application
    :PowerShell -Block:Test


    Will delete block data under Enterprise, Vendor, Application, Block specified.

    ComputerName                 Port            Operation          Status
    ------------                 ----            ---------          ------
    vproclient.vprodemo.com      16992           Delete             Successful
```

```
    -------------------------- EXAMPLE 5 --------------------------

    C:\PS>Get-Content computers.txt | Clear-AMT3PDS -TLS -Enterprise:Intel
-Vendor:Intel -Application:PowerShell -Block
    :Test


    ComputerName                Port            Operation          Status
    ------------                ----            ---------          ------
    computer1.vprodemo.com      16993           Delete             Successful
    computer2.vprodemo.com      16993           Delete             Successful
    computer3.vprodemo.com      16993           Delete             Successful
```

6.7 Intel AMT PowerShell GUI

The Intel AMT PowerShell Graphical User Interface (GUI) provides a simple interface for invoking most of the commands supported within the module.

6.7.1 Invoke-AMTGUI

NAME

Invoke-AMTGUI

SYNOPSIS

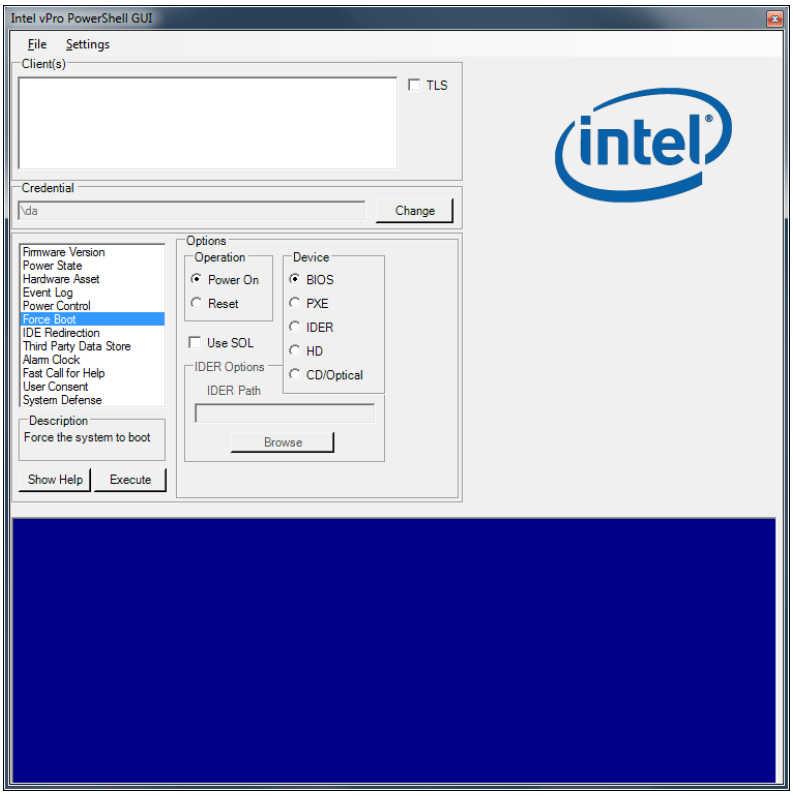
GUI that invokes PowerShell Module for Intel® vPro™ technology cmdlets

SYNTAX

Invoke-AMTGUI [[-ComputerName] <String[]>] [-Credential <PSCredential>][-xmlConfig] <string> [<CommonParameters>]

DESCRIPTION

The Intel AMT PowerShell Graphical User Interface (GUI) provides a simple interface for invoking most of the commands supported within the module. An xml configuration file can be passed in to configure the GUI. See the default XML in the invoke-amtgui.ps1 file.



.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Invoke-AMTGUI –Full

```
-------------------------- EXAMPLE 1 --------------------------

C:\PS>Invoke-AMTGUI


-------------------------- EXAMPLE 2 --------------------------
C:\PS>Invoke-AMTGUI $computerName


-------------------------- EXAMPLE 3 --------------------------
C:\PS>Invoke-AMTGUI $computerName -Credential $AmtCredential


-------------------------- EXAMPLE 4 --------------------------
C:\PS>Invoke-AMTGUI -xmlConfig sample.xml
```

6.8 Intel AMT User Consent

Intel AMT User Consent is a method of requesting consent from a user to remotely manage the client. To enforce the user's consent to opt-in for a redirection session, a secure output window (a "sprite") is displayed on the user's screen on top of any other window. The user is prompted to read out a randomly-generated number to the IT administrator. The redirection session is allowed to begin only if the IT administrator types in the correct number. Once a valid KVM Remote Control session is invoked, the client's entire screen is surrounded by a red bar, indicating that an IT administrator is in the process of a KVM Remote Control session.

6.8.1 Get-AMTUserConsent

NAME

Get-AMTUserConsent

SYNOPSIS

Gets the Intel AMT user consent state

SYNTAX

Get-AMTUserConsent [-ComputerName] <String[]> [[-Port] <String>] [-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet retrieves the Intel AMT user consent state from clients that have Intel AMT firmware version 3.0 and higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTUserConsent –Full

```
    -------------------------- EXAMPLE 1 --------------------------

    C:\PS>Write-AmtCredential

    -------------------------- EXAMPLE 2 --------------------------

    C:\PS>$AMTCredential = Write-AmtCredential  (will assume the digest account
"admin")

    -------------------------- EXAMPLE 3 --------------------------

    C:\PS>$AMTCredential = Get-Credential


    Write-AmtCredential –Username $AMTCredential.Username –Password
$AMTCredential.Password
```

6.8.2 Start-AMTUserConsent

NAME

Start-AMTUserConsent

SYNOPSIS

Starts the Intel AMT user consent process

SYNTAX

Start-AMTUserConsent [-ComputerName] <String[]> [-Port] <String> [-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet starts the user consent process on clients that have Intel AMT firmware version 3.0 and higher. The user consent screen is displayed on the remote client and the code must be passed into this cmdlet.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Start-AMTUserConsent –Full

```
-------------------------- EXAMPLE 1 --------------------------

C:\PS>Start-AMTUserConsent vproclient.vprodemo.com -credential $AMTCredential
```

6.8.3 Stop-AMTUserConsent

NAME

Stop-AMTUserConsent

SYNOPSIS

Stops the Intel AMT user consent process

SYNTAX

Stop-AMTUserConsent [-ComputerName] <String[]> [-Port] <String> [-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet stops the user consent process on clients that have Intel AMT firmware version 3.0 and higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Stop-AMTUserConsent –Full

```
------------------------- EXAMPLE 1 -------------------------

C:\PS>Stop-AMTUserConsent vproclient.vprodemo.com -credential $AMTCredential

ComputerName                Port                            Status
------------                ----                            ------
vproclient.vprodemo.com     16992                           Successful
```

6.9 Intel AMT IDER

Intel AMT IDE Redirection (IDER) is a technology that allows redirecting the floppy disk (IMG) or CD-ROM (ISO) from the console to a remote client. This client can then be booted from an ISO or IMG file for management or remediation.

6.9.1 Get-AMTIDER

NAME

Get-AMTIDER

SYNOPSIS

Lists the Intel AMT IDE Redirection sessions

SYNTAX

Get-AMTIDER [<CommonParameters>]

DESCRIPTION

This cmdlet lists the Intel AMT IDE redirection (IDER) sessions.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTIDER –Full

```
------------------------- EXAMPLE 1 -------------------------

C:\PS>Get-AMTIDER

ComputerName               IDERSessionID IDERPath         IDERState
------------               ------------- --------         ---------
192.168.1.100                          1 boot.iso          Disabled
```

6.9.2 Start-AMTIDER

NAME

Start-AMTIDER

SYNOPSIS

Starts an Intel AMT IDE redirection session

SYNTAX

Start-AMTIDER [-ComputerName] <String[]> [-Operation] <String> [[-IDERPath] <String>] [-TLS] [-AcceptSelfSignedCert] [-Username <String>] [-Password <String>] [[-Credential] <PSCredential>] [<CommonParameters>]

The valid parameters for –Operation are {PowerOn, Reset}.

DESCRIPTION

This cmdlet starts an Intel AMT IDE redirection (IDER) session on clients that have Intel AMT firmware version 3.0 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Start-AMTIDER –Full

```
   ------------------------- EXAMPLE 1 -------------------------

   C:\PS>Start-AMTider -computername:192.168.1.100 -operation:Reset
-iderpath:.\boot.iso -credential:$AMTCredential

   ComputerName         IDERSessionID IDERPath                    Status
   ------------         ------------- --------                    ------
   192.168.1.100                    1 boot.iso                    Successful
```

6.9.3 Stop-AMTIDER

NAME

Stop-AMTIDER

SYNOPSIS

Stops a specified Intel AMT IDE redirection session

SYNTAX

Stop-AMTIDER [[-IDERSessionID] <String[]>] [-CloseAllSessions] [<CommonParameters>]

DESCRIPTION

Stops a specified Intel AMT IDE Redirection (IDER) session. If no ID is specified, the oldest IDER session is closed.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Stop-AMTIDER –Full

```
   ------------------------- EXAMPLE 1 -------------------------

   C:\PS>Stop-AMTider Stop-AMTider SessionID

   ------------------------- EXAMPLE 2 -------------------------

   C:\PS>Stop-AMTIDER
```

```
-------------------------- EXAMPLE 3 --------------------------
C:\PS>Stop-AMTIDER -CloseAllSessions

Shuts down all IDER sessions
```

6.10 Configuration Cmdlets

This section describes cmdlets that help configure a system or provide more information about a client's configuration state.

6.10.1 Get-AMTSetup

NAME

Get-AMTSetup

SYNOPSIS

Returns Intel AMT setup information

SYNTAX

Get-AMTSetup [<CommonParameters>]

DESCRIPTION

This cmdlet retrieves Intel AMT setup information from the local system.

NOTES

Intel ME device drivers need to be installed prior to invoking this cmdlet. This command should be run on the local system.

The cmdlet requires elevated administrator privileges.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTSetup –Full

```
    ------------------------- EXAMPLE 1 -------------------------

    C:\PS>Get-AmtSetup


    SetupStatus                   : Unconfigured
    CoreVersion                   : 7.0.0
    BiosVersion                   : ASNBCPT1.86C.0036.B00.1008051344
    RemoteConfigurationSupported  :
    ConfigurationNonce            : bEYmf32WD19zvRvnNIjOzKGTg0U=
    ClientControlSupported        : True
    HostConfigurationSupported    : True
    ActiveHashes                  : {742C3192E607E424EB4549542BE1BBC53E6174E2, 132D0
                                      D45534B6997CDB2D5C339E25576609B5CC6,
2796BAE63F1
                                      801E277261BA0D77770028F20EEE4,
D1EB23A46D17D68FD
                                      92564C2F1F1601764D8E349...}
    HostName                      : VPROCOMPUTER
    IPAddress                     : {192.168.1.100, fe80::84a1:ffcc:d82e:90d3}
    DNSDomain                     : vprodemo.com
    DHCPEnabled                   : True
    UUID                          : 88888888-8887-8888-8888-878888888888
```

6.10.2 Enter-AMTRemoteConfiguration

NAME

Enter-AMTRemoteConfiguration

SYNOPSIS

Enters a Remote Configuration Session with an Intel AMT enabled client

SYNTAX

Enter-AmtRemoteConfiguration [-Session] <PSSession> [-Certificate] <X509Certificate> [[-otp] <String>] [-Force] [<CommonParameters>]

Enter-AmtRemoteConfiguration [-ComputerName] <String> [-Certificate] <X509Certificate> [[-Credential] <PSCredential>] [[-otp] <String>] [-Force] [<CommonParameters>]

DESCRIPTION

The Enter-AmtRemoteConfiguration cmdlet starts an interactive configuration session with an unconfigured Intel AMT enabled client with firmware version 3.2 and higher.

A PSRemoting session is required in order to discover information about the device and start the configuration service. A remote WS-MAN session is then established with the device and remains the active session until Stop-AMTConfiguration is called.

NOTES

- Intel AMT Provisioning: Intel AMT must in an unconfigured or remoteStarted state.
- The Intel® MEI Driver must be installed and working on the target Intel® AMT system.
- Intel AMT must be enabled in the BIOS.
- Configuration Sessions are not supported over wireless connections.
- The Intel onboard Wired LAN must be connected.
- This command can only be used remotely.

$ConfigurationCertificate must be set to the thumbprint of the desired certificate in the directory Microsoft.PowerShell.Security\Certificate::CurrentUser\my

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Enter-AMTRemoteConfiguration –Full

```
    -------------------------- EXAMPLE 1 --------------------------

    C:\PS>Enter-AmtRemoteConfiguration vprocomputer.vprodemo.com
$ConfigurationCertificate
```

6.11 Informational Cmdlets

This section describes the cmdlets that communicate directly with the client's Intel AMT firmware to return information about the client system.

6.11.1 Get-AMTAccessMonitor

NAME

Get-AMTAccessMonitor

SYNOPSIS

Returns Intel AMT access events

SYNTAX

Get-AMTAccessMonitor [-ComputerName] <String[]> [-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet returns the Intel AMT access events from clients that have Intel AMT firmware version 3.2 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTAccessMonitor –Full

```
-------------------------- EXAMPLE 1 --------------------------

C:\PS>Get-AMTAccessMonitor -computername:vProClient


ComputerName : vProClient
TimeStamp    : 1/01/2011 11:46:55 PM
Message      : AMT Provisioning Started
Location     :
User         :




-------------------------- EXAMPLE 2 --------------------------

C:\PS>Get-AMTAccessMonitor vProClient -Credential $amtcred - TLS | Format-Table

ComputerName       TimeStamp              Message            Location      User
------------       ---------              -------            --------      ----
vProClient         1/01/2011 11:46:55 PM  AMT Provisioning Sta...
```

6.11.2 Get-AMTEventLog

NAME

Get-AMTEventLog

SYNOPSIS

Returns the Intel AMT event log

SYNTAX

Get-AMTEventLog [-ComputerName] <String[]> [-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet returns the Intel AMT event log from clients that have Intel AMT firmware version 3.2 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTEventLog –Full

```
-------------------------- EXAMPLE 1 --------------------------
C:\PS>Get-AMTEventLog -computername:vProClient


ComputerName : vProClient
Severity     : 8
TimeStamp    : 2010/01/01 7:28:20 AM
Source       : BIOS
Message      : Starting operating system boot process



-------------------------- EXAMPLE 2 --------------------------

C:\PS>Get-AMTEventLog -computername:vProClient | Format-Table

ComputerName     Severity    TimeStamp               Source     Message
------------     --------    ---------               ------     -------
vProClient              8     2010/01/01 7:28:20 AM   BIOS       Starting
```

6.11.3 Get-AMTFirmwareVersion

NAME

Get-AMTFirmwareVersion

SYNOPSIS

Returns the core Intel AMT firmware version

SYNTAX

Get-AMTFirmwareVersion [-ComputerName] <String[]> [-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet returns the Intel AMT core firmware version from clients that have Intel AMT firmware version 3.2 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTFirmwareVersion –Full

```
-------------------------- EXAMPLE 1 --------------------------

C:\PS>Get-AMTFirmwareVersion -computername:vProClient

ComputerName                Property                    Value
------------                --------                    -----
vProClient                  AMT FW Core Version         7.1.3.1053
```

6.11.4 Get-AMTHardwareAsset

NAME

Get-AMTHardwareAsset

SYNOPSIS

Shows hardware information about the system

SYNTAX

Get-AMTHardwareAsset [-ComputerName] <String[]> [[-Port] <String>] [-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert] [-TextOutput] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet returns the hardware information from clients that have Intel AMT firmware version 3.2 or higher.

Use the **-TextOutput** switch to show the data in a text-only tree format.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTHardwareAsset –Full

```
-------------------------- EXAMPLE 1 --------------------------

C:\PS>get-AMTHardwareAsset vProClient.vprodemo.com -Credential $AMTCredential
```

```
    ComputerName                 PSParentPath                 Name
Value
    ------------                 ------------                 ----
-----
    vProClient.vprodemo.com      \HardwareAssets\BIOS\Prima... Version
1.16
    vProClient.vprodemo.com      \HardwareAssets\BIOS\Prima... ReleaseDate
2010/10/21 12:00:00 AM
    vProClient.vprodemo.com      \HardwareAssets\BIOS\Prima... Manufacturer
American Megatrends Inc.
    vProClient.vprodemo.com      \HardwareAssets\Cpu\CPU 0     CPUStatus
CPU Enabled
    vProClient.vprodemo.com      \HardwareAssets\Cpu\CPU 0     CurrentClockSpeed
2700
    vProClient.vprodemo.com      \HardwareAssets\Cpu\CPU 0
ExternalBusClockSpeed        100
    vProClient.vprodemo.com      \HardwareAssets\Cpu\CPU 0     Family
Intel(R) Core(TM) i7 proc
    e...
    vProClient.vprodemo.com      \HardwareAssets\Cpu\CPU 0     MaxClockSpeed
2700
    vProClient.vprodemo.com      \HardwareAssets\Cpu\CPU 0     UpgradeMethod
ZIF Socket
    vProClient.vprodemo.com      \HardwareAssets\Cpu\CPU 0     Manufacturer
Intel(R) Corporation
    vProClient.vprodemo.com      \HardwareAssets\Cpu\CPU 0     Version
Intel(R) Core(TM) i7-2620
    M...




    -------------------------- EXAMPLE 2 -------------------------

    C:\PS>Get-AMTHardwareAsset -ComputerName vProClient.vProDemo.com -Credential
$AMTCredential | Where-Object -FilterS
    c#ript {$_.PSPath -like "*BIOS*"}


    ComputerName                 PSParentPath                 Name
Value
    ------------                 ------------                 ----
-----
    vProClient.vprodemo.com      \HardwareAssets\BIOS\Prima... Version
1.16
    vProClient.vprodemo.com      \HardwareAssets\BIOS\Prima... ReleaseDate
2010/10/21 12:00:00 AM
    vProClient.vprodemo.com      \HardwareAssets\BIOS\Prima... Manufacturer
American Megatrends Inc.

    Displays only the results that contain "BIOS" in their path




    -------------------------- EXAMPLE 3 -------------------------

    C:\PS>Get-AMTHardwareAsset -ComputerName vProClient.vProDemo.com -Credential
$AMTCredential | Where-Object -FilterS
    c#ript {$_.PSPath -like "*BIOS*"} | format-list


    ComputerName : vProClient.vprodemo.com
    PSParentPath : AmtSystem::\HardwareAssets\BIOS\Primary BIOS
    Name         : Version
    Value        : 1.16

    ComputerName : vProClient.vprodemo.com
```

```
    PSParentPath : AmtSystem::\HardwareAssets\BIOS\Primary BIOS
    Name         : ReleaseDate
    Value        : 2010/10/21 12:00:00 AM

    ComputerName : vProClient.vprodemo.com
    PSParentPath : AmtSystem::\HardwareAssets\BIOS\Primary BIOS
    Name         : Manufacturer
    Value        : American Megatrends Inc.

    Displays only the results that contain "BIOS" in their path and formatted into
a list.




    -------------------------- EXAMPLE 4 --------------------------

    C:\PS>Get-AMTHardwareAsset -ComputerName vProClient -Credential $AMTCredential
-TextOutput


    vProClient BIOS
    vProClient  BIOS:Primary BIOS
         Version.............. 1.16
         ReleaseDate......... 2010/10/21 12:00:00 AM
         Manufacturer........ American Megatrends Inc.
    vProClient Cpu
    vProClient  Cpu:CPU 0
         CPUStatus........... CPU Enabled
         CurrentClockSpeed... 2700
         ExternalBusClockSpeed 100
         Family.............. Intel(R) Core(TM) i7 processor
         MaxClockSpeed....... 2700
         UpgradeMethod....... ZIF Socket
         Manufacturer........ Intel(R) Corporation
         Version.............. Intel(R) Core(TM) i7-2620M CPU @ 2.70GHz

    Displays results formatted as text.
```

6.11.5 Get-AMTPowerState

NAME

Get-AMTPowerState

SYNOPSIS

Returns the system power state

SYNTAX

Get-AMTPowerState [-ComputerName] <String[]> [[-Port] <String>] [-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert] [[-Credential] <PSCredential>] [<CommonParameters>]

DESCRIPTION

This cmdlet returns the system power state from clients that have Intel AMT firmware version 3.2 or higher.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTPowerState –Full

```
---------------------- Example 1 ----------------------

PS C:\Users\Administrator> Get-AMTPowerState -ComputerName dut.vprodemo.com -Username admin -Password Admin!98


Computer Name                  : dut.vprodemo.com
Power State ID                 : 2
Power State Description        : On (S0)
Power Saving State ID          : 3
Power Saving State Description : Power Saving
```

```
---------------------- Example 2 ----------------------

PS C:\Users\Administrator> Get-AMTPowerState -ComputerName dut.vprodemo.com -Username admin -Password Admin!98 -TLS


Computer Name                  : dut.vprodemo.com
Power State ID                 : 2
Power State Description        : On (S0)
Power Saving State ID          : 2
Power Saving State Description : Full Power
```

6.12 Intel Fast Call for Help

This section describes the cmdlets that allow configuration of an environment that supports Intel Fast Call for Help. Once the MPS proxies have been set up by using set AMTMPS, clients can be added to the MPS interface with set-AMTMPSClient. Afterwards all AMT cmdlets will transparently route to the client through the MPS interface.

- Set up proxy information with set-AMTMPS
- Identify when client connects to MPS.
- Add client with set-AMTMPSClient
- Call cmdlets with no change.

6.12.1 Get-AMTMPSStatus

NAME

Get-AMTMPSStatus

SYNOPSIS

Returns the status of the Intel Fast Call for Help Management Presence Server (MPS) interface settings

SYNTAX

Get-AMTMPSStatus [<CommonParameters>]

DESCRIPTION

Returns the status of the Intel Fast Call for Help Management Presence Server (MPS) interface settings

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-AMTMPSStatus –Full

```
-------------------------- EXAMPLE 1 --------------------------
get-AMTMPSStatus

HTTPProxy            SOCKSProxy            Client      Enabled
---------            ----------            ------      -------
HTTP Proxy address   SOCKS Proxy address                  True
```

6.12.2 Set-AMTMPS

NAME

Set-AMTMPS

SYNOPSIS

Set proxy information for the Intel Fast Call for Help Management Presence Server (MPS) interface

SYNTAX

Set-AMTMPS [-HTTPProxy] <String[]> [-SOCKSProxy] <String[]> [<CommonParameters>]

DESCRIPTION

Set proxy information for the Intel Fast Call for Help Management Presence Server (MPS) interface

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Set-AMTMPS –Full

```
-------------------------- EXAMPLE 1 --------------------------
set-AMTMPS -HTTPProxy HTTPProxy -SOCKSProxy SocksProxy
```

```
HTTPProxy          SOCKSProxy          Client        Enabled
---------          ----------          ------        -------
HTTP Proxy address SOCKS Proxy address                  True
```

6.12.3 Set-AMTMPSClient

NAME

Set-AMTMPSClient

SYNOPSIS

Add and remove clients from the Intel Fast Call for Help Management Presence Server (MPS) interface

SYNTAX

Set-AMTMPSClient [-action] <String[]> [-hostname] <String[]> [<CommonParameters>]

The valid parameters for **–action** are {add, remove}

DESCRIPTION

Add and remove clients from the Intel Fast Call for Help Management Presence Server (MPS) interface

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Set-AMTMPSClient –Full

```
-------------------------- EXAMPLE 1 --------------------------
set-AMTMPSClient -action add -hostname DemoClient

Added DemoClient to MPS client list


-------------------------- EXAMPLE 2 --------------------------
set-AMTMPSClient -action remove -hostname DemoClient

Removed DemoClient from MPS client list
```

6.12.4 Clear-AMTMPS

NAME

Clear-AMTMPS

SYNOPSIS

Clears the Intel Fast Call for Help Management Presence Server (MPS) interface settings

SYNTAX

Clear-AMTMPS [<CommonParameters>]

DESCRIPTION

Clears the Intel Fast Call for Help Management Presence Server (MPS) interface settings

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Clear-AMTMPS –Full

```
-------------------------- EXAMPLE 1 --------------------------
clear-AMTMPS

MPS settings cleared
```

6.13 Intel® UPID

6.13.1 Get-UniquePlatformIDFeatureInfo

NAME

Get-UniquePlatformIDFeatureInfo

SYNOPSIS

Returns Intel® Unique Platform ID feature information

SYNTAX

Get-UniquePlatformIDFeatureInfo

DESCRIPTION

This CmdLet invokes Intel MEI WMI commands to get the Unique Platform ID feature information.

NOTES

- Intel ME device drivers need to be installed prior to invoking this cmdlet.

  If your computer is missing the Intel CSME WMI provider (which is usually pushed by Microsoft Windows Update), you need to install the Intel CSME Software package, which includes the provider. The Intel CSME Software package and installation instructions are available from your OEM. Alternatively, you can download the Intel CSME Software from Intel's Download Center.
- This command should be run on the local system.

For more details, review the Windows PowerShell integrated help by typing:

Get-Help Get-UniquePlatformIDFeatureInfo –Full

```
------------------------- Example 1 -------------------------

PS C:\Users\Administrator> Get-UniquePlatformIDFeatureInfo


LMSservice started   : True
UPID state           : True
OEM Platform ID Type : 0
OEM Platform ID      : 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
                       0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
CSME Platform ID     : 0x9b 0x4b 0xa3 0xb5 0xe3 0x46 0xe9 0x80 0x1c 0x3e 0xc7 0x70 0x83 0x37 0x44 0x6a 0xe1 0x68 0xbf
                       0xf6 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x82 0x51 0x00 0x00
```

6.14 TLS Configuration Flow Using PowerShell Snippets

6.14.1 Create Intel AMT Certificate and Key for TLS Connection

The following flow describes how to create an Intel AMT certificate and key to enable connecting over TLS.

Step 1:

Invoke-GenerateKeyPair

Run the **Invoke-GenerateKeyPair** cmdlet (see *Cmdlets for TLS Configuration*). This command will generate a new key and add it to the Intel AMT certificate store.

Note

This cmdlet includes this line (commented-out):

```
$keyBlob = $publicPrivateKeyPairInstance.GetProperty("DERKey")
```

The key pair data is needed later in the flow. Please note that this key pair data is sensitive and should be kept secret, so make sure you protect it properly in your implementation.

Step 2:

Create a null signed certificate request based on the key generated in Step 1.
This should be done using 3rd party software.

Step 3:

Invoke-GenerateCSR

Run the **Invoke-GenerateCSR** cmdlet (see *Cmdlets for TLS Configuration*).

Input for this cmdlet:

- Path to the null signed certificate request created in Step 2.

Note

This cmdlet includes this line (commented-out):

```
$signedCertificateRequest =
$outputObject.GetProperty("SignedCertificateRequest")
```

The signed certificate request is needed later in the flow. Please note that this signed certificate request data is sensitive and should be kept secret, so pay attention to properly protect it in your implementation.

Step 4:

Create the Intel AMT certificate based on the generated signing certificate request from Step 3.
This should be done using 3rd party software.

6.14.2 Enable TLS Server Authentication Connection

Invoke-ConfigureTLSServerAuthentication

Prerequisites:

- The Intel AMT private key should be added already (using the procedure in *Create Intel AMT Certificate and Key for TLS* Connection or *Additional TLS Scripts*).
- The client should hold the root CA in its certificate store.

Run the Invoke-ConfigureTLSServerAuthentication cmdlet (see *Cmdlets for TLS Configuration*).

Input for this cmdlet:

- Path to the Intel AMT certificate file. (If the procedure in *Create Intel AMT Certificate and Key for TLS* Connection was performed, use the certificate created in Step 4 of that procedure.)

6.14.3 Enable TLS Mutual Authentication Connection

Invoke-ConfigureTLSMutualAuthentication

Prerequisites:

- The Intel AMT private key should be added already (using the procedure in *Create Intel AMT Certificate and Key for TLS Connection* or *Additional TLS Scripts*).
- The client certificate should be installed in the client certificate store.
- The client should hold the root CA in its certificate store.
- Make sure the Intel AMT system time is synchronized with the remote machine.

Run the **Invoke-ConfigureTLSMutualAuthentication** cmdlet (see *Cmdlets for TLS Configuration*).

Input for this cmdlet:

- Path to the Intel AMT certificate file. (If the procedure in *Create Intel AMT Certificate and Key for TLS* Connection was performed, use the certificate created in Step 4 of that procedure.)
- Path to the client's trusted root CA certificate file

6.14.4 Disable TLS Authentication

Invoke-DisableTLSAuthentication

Run the **Invoke-DisableTLSAuthentication** cmdlet (see *Cmdlets for TLS Configuration*).

If Intel AMT is configured for TLS-only connection, use the **-TLS** parameter to invoke this command.
For mutual authentication only, use also the **-CertificateName** parameter.

6.14.5 Additional TLS Scripts

Invoke-AddPrivateKey

Run the **Invoke-AddPrivateKey** cmdlet (see *Cmdlets for TLS Configuration*).

This script should be used if you have an already created key file that you want to add to Intel AMT.

Input for this cmdlet:

- Path to the key file.

6.14.6 Cmdlets for TLS Configuration

6.14.6.1 Invoke-GenerateKeyPair

NAME
Invoke-GenerateKeyPair

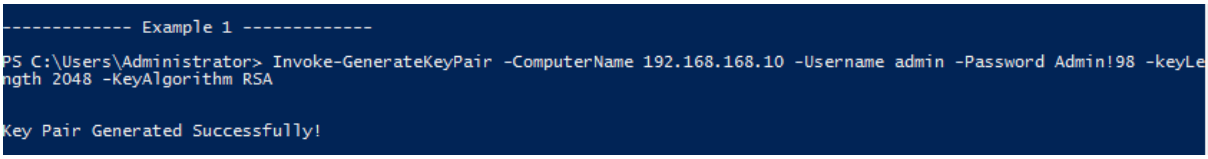
SYNOPSIS
Generate key Pair.

SYNTAX

```
Invoke-GenerateKeyPair [-ComputerName <String[]>] [-Port <String>] [-KeyAlgorithm
<String>]
[-KeyLength <String>] [-TLS] [-AcceptSelfSignedCert][-Username <String>]
[-Password <String>][[-Credential] <PSCredential>] [<CommonParameters>]
```

The valid parameter value for **–KeyAlgorithm** is {RSA}.

DESCRIPTION
This cmdlet generates a KeyPair object and adds the new key to the Intel AMT key store.



6.14.6.2 Invoke-GenerateCSR

NAME
Invoke-GenerateCSR

SYNOPSIS
Generates a certificate signing request based on a key from the key store.

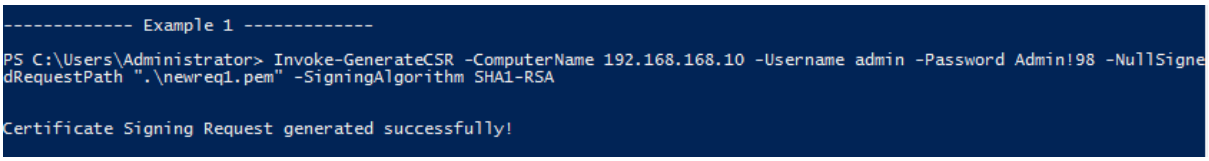
SYNTAX

```
Invoke-GenerateCSR [-ComputerName] <String[]> [-Port] <String> [-SigningAlgorithm
<String>]
[-NullSignedRequestPath <String>] [-TLS] [-AcceptSelfSignedCert]
[-Username <String>] [-Password <String>][[-Credential] <PSCredential>]
[<CommonParameters>]
```

The valid parameter value for **-NullSignedRequestPath** is a path to a null signed certificate request file based on the Intel AMT generated key. This file should be provided by the user.

The valid parameter values for **–SigningAlgorithm** are {SHA1-RSA, SHA256-RSA}.

DESCRIPTION
This cmdlet generates a certificate signing request based on a key from the key store.



6.14.6.3 Invoke-ConfigureTLSServerAuthentication

NAME

Invoke-ConfigureTLSServerAuthentication

SYNOPSIS

Configures AMT connection to TLS with server authentication.

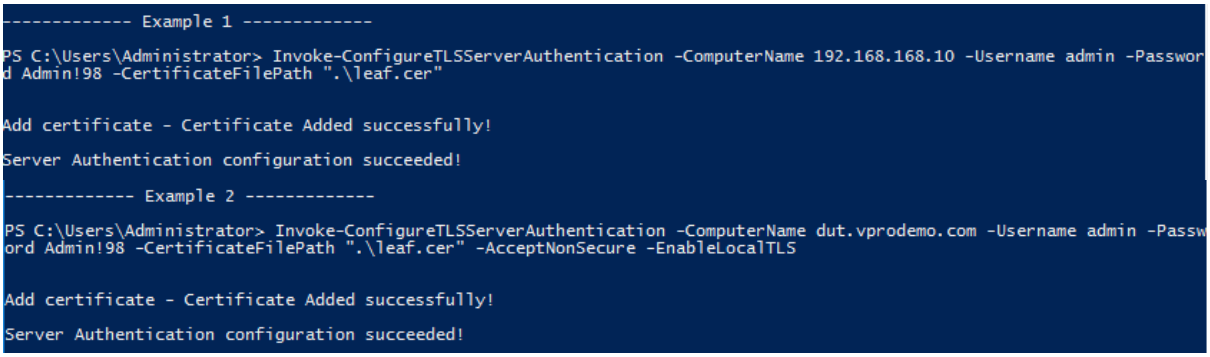
SYNTAX

```
Invoke-ConfigureTLSServerAuthentication [-ComputerName] <String[]> [-Port]
<String> [-CertificateFilePath <String>]
[-AcceptNonSecure ] [-EnableLocalTLS] [-TLS] [-AcceptSelfSignedCert] [-Username
<String>] [-Password <String>][[-Credential] <PSCredential>]
[<CommonParameters>]
```

The parameter for -CertificateFilePath is a path to a certificate file.

DESCRIPTION

This cmdlet configures the Intel AMT connection to TLS with server authentication.
The cmdlet adds the Intel AMT certificate and enables TLS.
Use the **-AcceptNonSecure** parameter to allow also non secure connection.
Use the **-EnableLocalTLS** parameter to enable local TLS connection in addition to the remote connection.



6.14.6.4 Invoke-ConfigureTLSMutualAuthentication

NAME

Invoke-ConfigureTLSMutualAuthentication

SYNOPSIS

Configures Intel AMT connection to TLS with mutual authentication.

SYNTAX

```
Invoke-ConfigureTLSMutualAuthentication [-ComputerName] <String[]> [-
Port] <String> [-CertificateFilePath <String>]
[-TrustedRootCertificateFilePath <String>] [-AcceptNonSecure ] [-TLS]
[-AcceptSelfSignedCert] [-Username <String>] [-Password <String>] [[-
Credential] <PSCredential>] [<CommonParameters>]
```

The parameter for **-CertificateFilePath** is a path to a certificate file.
The parameter for **-TrustedRootCertificateFilePath** is a path to a certificate file that is a trusted root certificate file.

DESCRIPTION

This cmdlet configures the Intel AMT connection to TLS with mutual authentication.
The cmdlet adds the Intel AMT certificate, client certificate and enables TLS.
Use the **-AcceptNonSecure** parameter to allow also non-secure connection.

```
-------------- Example 1 --------------

PS C:\Users\Administrator>Invoke-ConfigureTLSMutualAuthentication -ComputerName 192.168.168.10 -Username admin -Password
 Admin!98 -CertificateFilePath ".\leaf.cer" -TrustedRootCertificateFilePath ".\trusted_cert.cer"


Add public key certificate - certificate added successfully!

Add trusted root certificate - certificate added successfully!

Mutual Authentication configuration succeeded!
```

```
-------------- Example 2 --------------

PS C:\Users\Administrator>Invoke-ConfigureTLSMutualAuthentication -ComputerName dut.vprodemo.com -Username admin -Passwo
rd Admin!98 -CertificateFilePath ".\leaf.cer" -TrustedRootCertificateFilePath ".\trusted_cert.cer" -AcceptNonSecure


Add public key certificate - certificate added successfully!

Add trusted root certificate - certificate added successfully!

Mutual Authentication configuration succeeded!
```

6.14.6.5 Invoke-DisableTLSAuthentication

NAME

Invoke-DisableTLSAuthentication

SYNOPSIS

Disables the TLS connection in Intel AMT.

SYNTAX

```
Invoke-DisableTLSAuthentication [-ComputerName] <String[]> [-Port]
<String>
[-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert]
[-CertificateName <String>]
[-Credential] <PSCredential>] [<CommonParameters>]
```

DESCRIPTION

This cmdlet disables the TLS connection (server or mutual) in Intel AMT.

```
------------ Example 1 ------------

PS C:\Users\Administrator> Invoke-DisableTLSAuthentication -ComputerName dut.vprodemo.com -Username admin -Password Admi
n!98


Disable TLS authentication succeeded!

------------ Example 2 ------------

PS C:\Users\Administrator> Invoke-DisableTLSAuthentication -ComputerName dut.vprodemo.com -Username admin -Password Admi
n!98 -TLS -CertificateName "management_console.VproDemo.com"


Disable TLS authentication succeeded!
```

6.14.6.6 Invoke-AddPrivateKey

NAME
Invoke-AddPrivateKey

SYNOPSIS
Add a private key to Intel AMT.

SYNTAX

```
Invoke-AddPrivateKey [-ComputerName] <String[]> [-Port] <String> [-
PrivateKeyPath <String>]
[-Username <String>] [-Password <String>] [-TLS] [-AcceptSelfSignedCert]
[-CertificateName <String>] [-Credential] <PSCredential>]
[<CommonParameters>]
```

The parameter for **-PrivateKeyPath** is a path to a key file.

DESCRIPTION
This cmdlet adds a private key to Intel AMT using a private key file provided by the user.

```
--------------- Example 1 ---------------

PS C:\Users\Administrator> Invoke-AddPrivateKey -ComputerName 192.168.168.10 -Username admin -Password Admin!98 -private
KeyPath ".\privateKey.pem"


Key Added successfully!
```

6.15 OCR (One Click Recovery) Cmdlet

6.15.1.1 Invoke-AMTForceBoot_OCR

NAME
Invoke-AMTForceBoot_OCR

SYNOPSIS
Invokes the Intel Active Management Technology force boot command using the One-Click Recovery boot options.

SYNTAX

```
Invoke-AMTForceBoot [-ComputerName] <String[]> [[-Port] <String>] [-TLS]
[-AcceptSelfSignedCert]  [-BootSource]<String> [-ParametersArray_path]
<String> [-NumberOfParameters] <String> [-Username <String>] [-Password
<String>][[-Credential] <PSCredential>] [<CommonParameters>]
```

The valid parameter value for **-ParameterArray_path** is a path to a file that contains the UEFI boot parameter array encoded to base64. This parameter is optional.

The valid parameter value for **-NumberOfParameters** is the number of parameters included in the UEFI boot parameter array. This parameter is required only if the **-ParameterArray_path** parameter is set.

The valid parameter values for **–BootSource** are {HTTPS, PBA, WinRE}.

- **HTTPS** – Forces a boot to an HTTPS server
- **PBA –** Forces a boot to a locally-installed recovery or diagnostics pre-boot application
- **WinRE –** Forces boot to Microsoft Windows RE

The **-TLS** parameter is mandatory for this cmdlet.

DESCRIPTION

This cmdlet invokes the Intel Active Management Technology force boot command using the One-Click Recovery boot options.
The available OCR boot options are: HTTPS server, PBA, and WinRE.

The OCR feature is available on vPro SKUs, starting with Tiger Lake platforms using Intel CSME 15 firmware.

This cmdlet does the following:

1. Checks whether the selected OCR boot capability is supported.
2. Checks whether the selected OCR boot capability is enabled.
3. Displays available OCR sources for the selected option.
4. Sets the next boot to an OCR selected boot source.
5. Performs remote-control reset / power on, to trigger the OCR boot.

NOTE

The OCR commands are supported only over TLS connection. For running this cmdlet, the **-TLS parameter** is mandatory.

Figure 3 - WinRE boot option example

Figure 4 - HTTPS server boot option example

7 AMTSystem PowerShell Drive Provider

Microsoft has added the concept of a Windows PowerShell drive to Windows PowerShell version 2.0. These drives are information stores that can be accessed like a file system drive. Many drives are created automatically, such as the Registry (HKCU: and HKLM:), the certificate store (Cert:) and the Environment (ENV:).

For information on Windows PowerShell drives, see https://docs.microsoft.com/en-us/powershell/scripting/samples/managing-windows-powershell-drives?view=powershell-6.

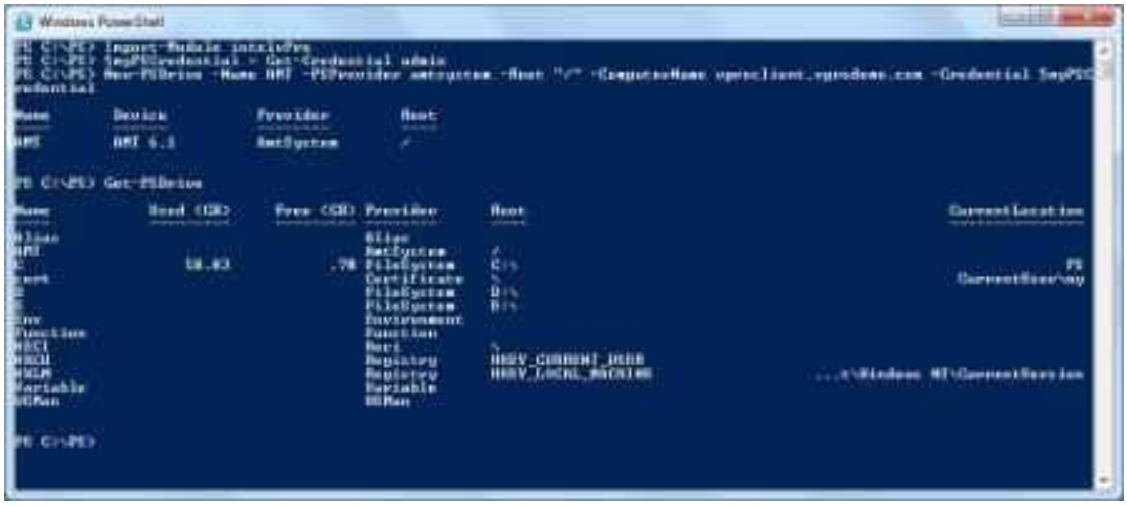
Intel has created a PowerShell Drive provider called AMTSystem that allows a remote Intel vPro technology enabled client to be accessed like a drive. This powerful feature allows the remote system's Intel vPro technology settings to be easily listed, accessed and changed.

You can map a PowerShell drive called "AMT" to a remote system with Intel vPro technology. To do so, run the following command from the PowerShell console:

New-PSDrive -Name AMT -PSProvider amtsystem -Root "\" –ComputerName vproclient.vprodemo.com -Credential $myPScredential

If your AMT client is configured in TLS mode (TLS encrypted traffic over AMT Port 16993), add the **–TLS** switch to the command.

Type **Get-PSDrive** to list the available drives.



Once the AMT PowerShell Drive has been mapped, you can browse and navigate to the remote system, similar to the way you navigate to a normal file system drive:

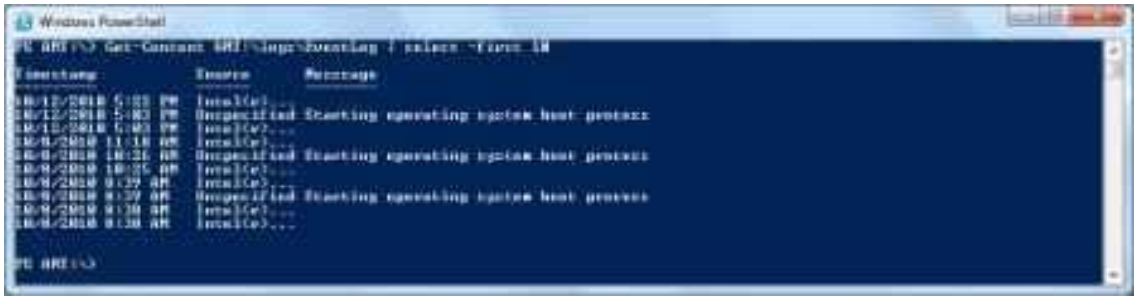
cd AMT:

dir

What can you do with this newly-mapped drive?

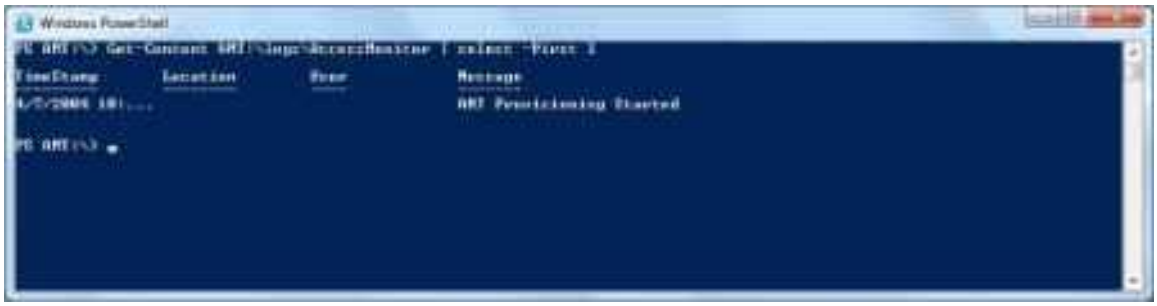
To display the AMT Event log:

Get-Content AMT:\logs\EventLog



To display the AMT Access Monitor (Audit Log):

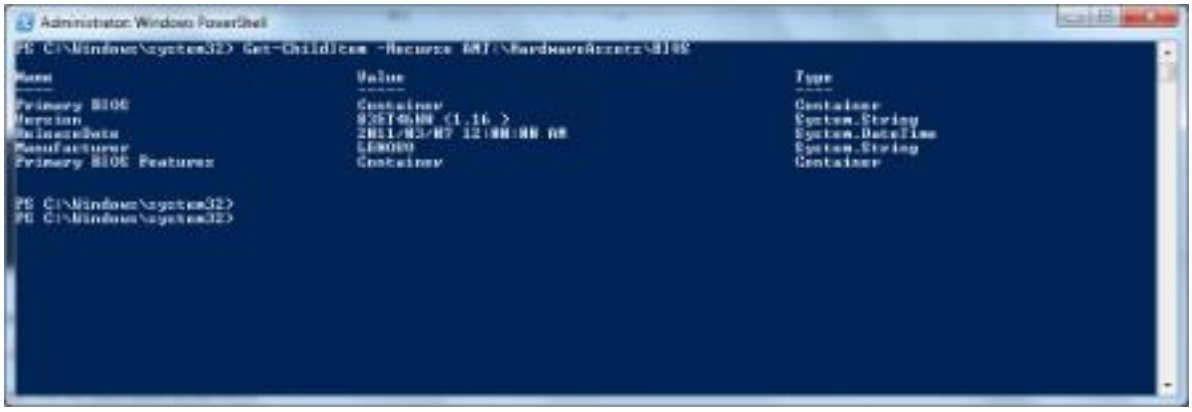
Get-Content AMT:\logs\AccessMonitor



You can enumerate the system Hardware Inventory and dump the data to a file for auditing purposes:

Get-ChildItem -Recurse AMT:\HardwareAssets | Out-File C:\PS\HWInv.txt

To reduce the amount of information and focus on the BIOS items only:
Get-ChildItem -Recurse AMT:\HardwareAssets\BIOS

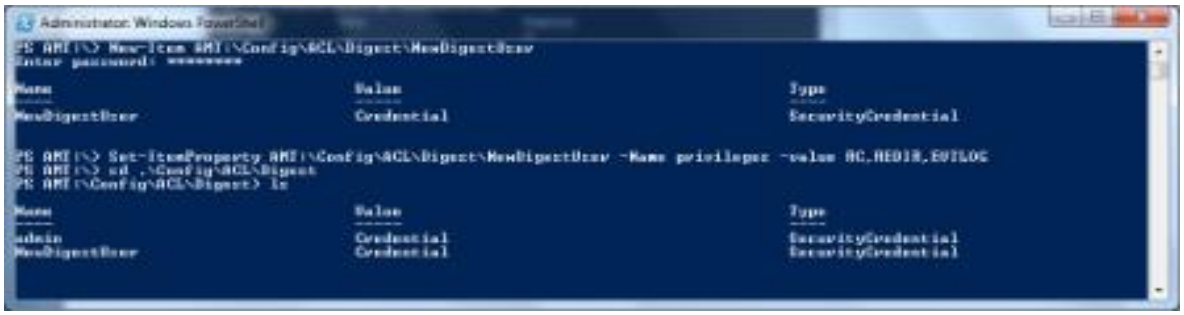


To turn on IDE-R:
Set-Item AMT:\Config\Redirection\IderEnabled -value "True"

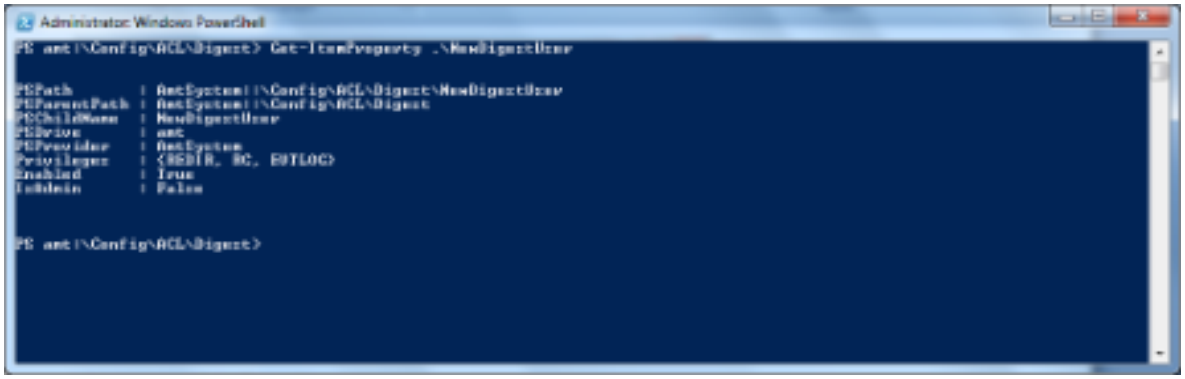
To turn off KVM User consent:
Set-Item AMT:\Config\KVM\UserConsent -value "False"

To change the AMT hostname:
Set-Item AMT:\Config\etc\Hosts\HostName "NewHostName"

To add a new user and give them rights:
New-Item AMT:\Config\ACL\Digest\NewDigestUser -Password P@ssw0rd
Set-ItemProperty AMT:\Config\ACL\Digest\NewDigestUser -Name Privileges -Value RC,REDIR,EVTLOG



To check the properties of the newly-added user:
Get-ItemProperty NewDigestUser



A Appendix A: QuickStart Guide

This appendix provides information to help you quickly install, set up and use the Intel vPro Technology Module for Microsoft Windows PowerShell.

A.1 Download the Module

Download the latest version of the module from https://software.intel.com/content/www/us/en/develop/download/intel-vpro-technology-module-for-microsoft-windows-powershell-module.html

A.2 Unzip the Module Package Folder

Unzip the contents of the folder on the system from which you want to run the module.

A.3 Set Execution Level

Open a PowerShell console as an administrator and type

Set-ExecutionPolicy RemoteSigned

A.4 Set Credentials

In the PowerShell console type

$AMTCred = get-credential

A.5 Run Cmdlets

Use cmdlets to manage the Intel vPro technology enabled client.

Get-AMTPowerState "ComputerName" –Credential $AMTCred

Invoke-AMTPowerManagement "ComputerName" –Credential $AMTCred –Operation PowerOn

Invoke-AMTGUI "ComputerName" –Credential $AMTCred

B Appendix B: General Cmdlet and Function Methodology

This appendix provides additional information on the methodology used in developing the cmdlets and functions for the Windows PowerShell Module for Intel vPro Technology.

B.1 Verb-Noun Pair Compliance

The Windows PowerShell Module for Intel vPro Technology actively complies with Windows PowerShell verb-noun pair convention for the names of cmdlets and functions. The verb part of the name identifies the action that the cmdlets and functions perform. The noun part of the name identifies the entity on which the action is performed. For more information on the Windows PowerShell verb-noun methodology, visit the following link:

http://msdn.microsoft.com/en-us/library/ms714428(VS.85).aspx

B.2 Cmdlet and Function Parameters

With every Windows PowerShell Module for Intel vPro Technology cmdlets and functions, there is a consistent set of parameters that all the cmdlets use.

Table 2: Cmdlet and Function Parameters

Parameter	Description	Positional Input	Pipeline Input	Required
ComputerName	Managed Client hostname, FQDN, IP address, or array of the previous	Yes	ByValue, ByPropertyName	True
TLS	Switch to specify if TLS should be used to communicate with the client. 16992 for non-TLS, 16993 for TLS. Default is 16992 if TLS is not specified.	No	ByPropertyName	False
Username	Digest or Kerberos User to authenticate with	No	ByPropertyName	False

Parameter	Description	Positional Input	Pipeline Input	Required
Password	Password for Digest or Kerberos User	No	ByPropertyName	False
Credential	Preferred mechanism for authentication using PS-Credential	Yes	ByPropertyName	False

The following parameters are consistently used (they may not exist in every cmdlet and function:

Parameter	Description	Positional Input	Pipeline Input	Required
Operation	Typically a sub operation of the cmdlets or functions	Yes	ByPropertyName	True

The Windows PowerShell Get-Help command can be used on the desired cmdlets and functions to list any additional optional or required parameters.

B.3 Cmdlets and Functions Integrated Help

Each Windows PowerShell Intel vPro Module cmdlet and function supports the **Get-Help** command. Following is an example of using the Get-Help command:

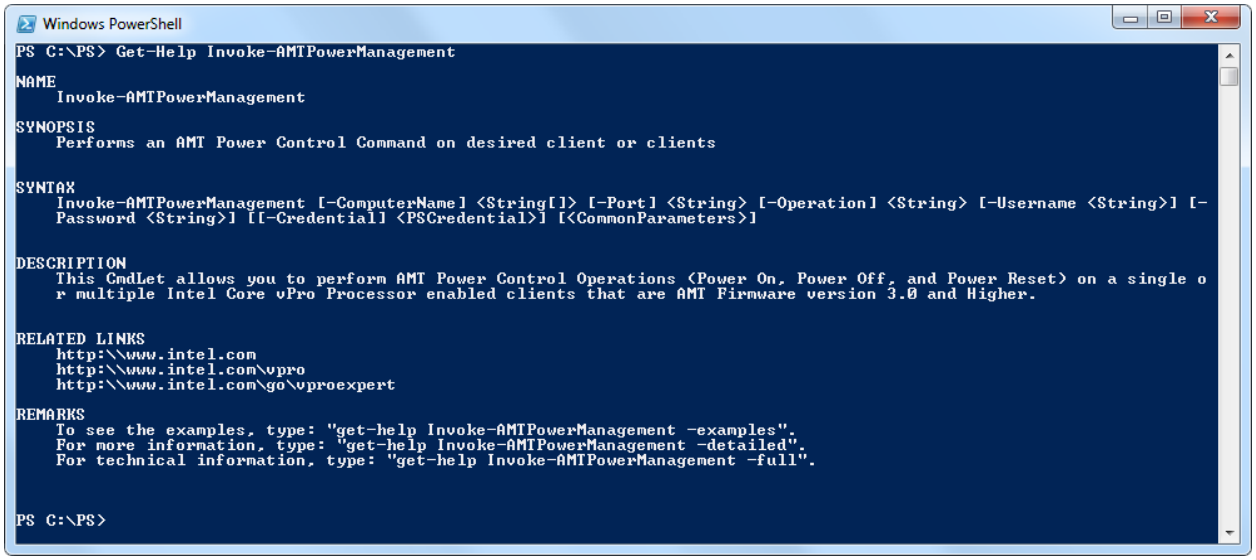


Figure 5: Module Help

You can display more detailed information on the cmdlet and function and how to use it by using the **Full**, **Detailed**, and **Examples** parameters with Get-Help.