Citrix Provisioning Services 7.15
PowerShell with Objects Programmer's Guide

Revision 1

August 2017

Copyright and Trademark Notice

Table of Contents

Introduction

Use Provisioning Services' programming interfaces to manage your implementation from a command line or from scripts. Only users with correct administrative privileges can use programming commands. Non-administrators, that do not have elevated privileges and attempt to use these commands, will receive the 'Invalid access' message.

Four different programming interfaces exist:

- Management Command Line Interface (MCLI)
- Simple Object Access Protocol (SOAP) Server Programmer Interface
- PowerShell Programmer Interface with Objects
- PowerShell Programmer Interface (Deprecated)

This document provides the information needed to use this interface.

Using the PowerShell Programmer Interface with Objects

Use the information that follows to manage a Provisioning Service's implementation from the PowerShell Interface with Objects.

Installation of PowerShell Snap-In

The PowerShell snap-in (Citrix.PVS.SnapIn.dll) can be installed using the Provisioning Server Console install.

Registration of McliPSSnapIn.dll using Import-Module

If the snap-in later needs to be registered in PowerShell, this can be manually done by running one of the following command in PowerShell. The path where the Citrix.PVS.SnapIn.dll is installed needs to be used.

Import-Module "*path*\Citrix.PVS.SnapIn.dll"

Registration of Citrix.PVS.SnapIn.dll using Add-PSSnapin

If the snap-in later needs to be registered in PowerShell, this can be manually done by running one of the following commands at the DOS command prompt. After the command is run and PowerShell is started, the snap-in needs to be added using the command "Add-PSSnapin -Name Citrix.PVS.SnapIn".

64-bit Registration

```
%systemroot%\Microsoft.NET\Framework64\v4.0.30319\installutil.exe
Citrix.PVS.SnapIn.dll
```

32-bit Registration

```
%systemroot%\Microsoft.NET\Framework\v4.0.30319\installutil.exe
Citrix.PVS.SnapIn.dll
```

Alternative Registration of Citrix.PVS.SnapIn.dll in PowerShell

Another way to register is by running one of the following commands at the PowerShell command prompt:

64-bit Registration in PowerShell

```
$installutil = $env:systemroot +
'\Microsoft.NET\Framework64\v4.0.30319\installutil.exe'

&$installutil Citrix.PVS.SnapIn.dll

Add-PSSnapin -Name Citrix.PVS.SnapIn
```

32-bit Registration in PowerShell

```
$installutil = $env:systemroot +
'\Microsoft.NET\Framework32\v4.0.30319\installutil.exe'

&$installutil Citrix.PVS.SnapIn.dll

Add-PSSnapin -Name Citrix.PVS.SnapIn
```

Uninstall of PowerShell Snap-In

The PowerShell snap-in (Citrix.PVS.SnapIn.dll) can be uninstalled using the Provisioning Server Console install.

Unregister of Citrix.PVS.SnapIn.dll

The snap-in needs to be unregistered from PowerShell, this can be manually done by running one of the following commands at the DOS command prompt:

64-bit Unregister

%systemroot%\Microsoft.NET\Framework64\v4.0.30319\installutil.exe -u Citrix.PVS.SnapIn.dll

32-bit Unregister

%systemroot%\Microsoft.NET\Framework\v4.0.30319\installutil.exe -u Citrix.PVS.SnapIn.dll

Alternative Unregister of Citrix.PVS.SnapIn.dll in PowerShell

Another way to unregister is by running one of the following commands at the PowerShell command prompt:

64-bit Unregister in PowerShell

```
$installutil = $env:systemroot +
'\Microsoft.NET\Framework64\v4.0.30319\installutil.exe'

&$installutil -u Citrix.PVS.SnapIn.dll
```

32-bit Unregister in PowerShell

```
$installutil = $env:systemroot +
'\Microsoft.NET\Framework32\v4.0.30319\installutil.exe'

&$installutil -u Citrix.PVS.SnapIn.dll
```

Setup of the SOAP Server Communication

Unless the defaults are fine, use this command to set the values for the SOAP Server connection:

Set-PvsConnection

```
Set the SoapServer connection, and if -Persist is specified the connection
  settings are saved in the registry. A PvsConnection object can be used as
  the parameter.

   Required

    PvsConnection Connection: PvsConnection object with changed property
                    value(s) to be set. The object can come from a
                    pileline.

    These values are in the PvsConnection object, and only will be set if
                    the value has changed.

    string Name or Server: Name or IP of the Server to connect to.
                    Default=localhost

    string Port: The Port to use to connect. Default=54321

    string User: User name to use for Authentication. If it has a value,
                    it will be *****. Default=""

    string Domain: Domain name to use for Authentication. If it has a
                    value, it will be *****. Default=""

    string Password: Password to use for Authentication. If it has a
                    value, it will be *****. Default=""

    string Persist: True when the connection settings should be, for Set,
                    or have been, for Get, saved to the registry.

   PvsConnection can be created or modified using methods below:

    New-Object Citrix.PVS.SnapIn.PvsConnection: Creates default
      Server=localhost, Port=54321, and no authentication.

    New-Object Citrix.PVS.SnapIn.PvsConnection(Citrix.PVS.SnapIn
      copyFrom): Creates with settings of the copyFrom Citrix.PVS.SnapIn.

    SetServerToLocalHostDefaultSettings: Server=localhost, Port=54321, and
      no authentication.

    Copy(Citrix.PVS.SnapIn copyFrom): Modifies the settings to match the
      copyFrom Citrix.PVS.SnapIn.

    Equals(Citrix.PVS.SnapIn compareTo): Returns true when the settings
      match what is in the compareTo.


   When Connection is not passed, the parameters below are used:
```

```
Optional field values to set:

 string Name or Server: Name or IP of the Server to connect to.
                 Default=localhost

 string Port: The Port to use to connect. Default=54321

 string User: User name to use for Authentication. If it has a value,
                 it will be *****. Default=""

 string Domain: Domain name to use for Authentication. If it has a
                 value, it will be *****. Default=""

 string Password: Password to use for Authentication. If it has a
                 value, it will be *****. Default=""

 string Persist: True when the connection settings should be, for Set,
                 or have been, for Get, saved to the registry.


Optional

 SwitchParameter PassThru: If -PassThru is specified, the resulting
                 PvsConnection object is returned.

 SwitchParameter Confirm: The impact of this operation is "low". If -
                 Confirm is specified, the operation will be
                 confirmed. $ConfirmPreference can be set to "low" to
                 have confirmation without the Confirm parameter.
```

EXAMPLE 1: Set PvsConnection for Individual Fields

```
 Get the PvsConnection into a $o variable. Change the $o field values
                 and then Set the PvsConnection with the result.

 $o = Get-PvsConnection -Fields Port

 $o.Port = 54322

 Set-PvsConnection $o

 The -Fields parameter with only the needed fields specified makes the
                 Get work faster because only those fields are
                 retrieved.
```

EXAMPLE 2: Set PvsConnection for a Field Using Pipe

```
 Get the PvsConnection into a $o variable for the field that has the
                 wrong value. Change the $o field to the correct value
                 and then Set the PvsConnection with the result.

 Get-PvsConnection -Fields Port | Where-Object {$_.Port -ne 54322} |
                 foreach { $o = $_; $o.Port = 54322; $o } | Set-
                 PvsConnection

 The -Fields parameter with only the needed fields specified makes the
                 Get work faster because only those fields are
                 retrieved.

 The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y
                 and returns the object again so it can be piped to
                 the Set command for update.
```

EXAMPLE 3: Set PvsConnection Port with Parameter

```
Set the PvsConnection Port using the Port parameter instead of a
                PvsConnection object.

Set-PvsConnection -Port 54322

This is the only Set command that has field parameters.
```

Command Specific Help

All of the documentation for specific commands is included in the command-line help by calling Get-Help and the name of the command, for example “Get-Help Get-PvsDevice”. The documentation for the specific command includes information about the object that is may get or set. The documentation of all of the objects is included in the “Objects, in the Citrix.PVS.SnapIn Namespace” section.

Error Handling

For the Citrix.PVS.SnapIn, if an error occurs, a PvsException will be in the Exception member of the $error. The “Error codes” information is included in this document.

The members of a PvsException are:

InnerException: The exception that occured. This exception maybe an EAException or other standard Exception.

ToString(): Has the formatted full Message of the InnerException.

If the InnerException GetType().Name equals "EAException", then The members of it are:

returnCode: The number, as shown below in the Error codes. The name of the error, for example "NotImplemented", is not included in the EAException.

Message: The message, as shown below in the Error codes. The [v1], [v2], [v3], [v4], and [v5] will be replaced with values as required.

Details: Has the Details for the EAException if there are any. OtherException, ManagementInterfaceError and PvsStatusException will have Details.

ToString(): Has the Message as shown below in the Error codes. If there is Details, it will be returned or included, and if partialReturn, they will be included.

partialReturn: Might have a list of EAException objects if any of the items processed during the command had any issues.

Severity: Can have the values Critical, Error, Warning or Duplicate.

Source: Has the value that is displayed in the Console as a Title or Type for the error.

Short cmdlet list

Add-PvsDeviceToDomain

Add-PvsDeviceToView

Add-PvsDiskLocatorToDevice

Add-PvsDiskToUpdateTask

Add-PvsDiskVersion

Clear-PvsConnection

Clear-PvsTask

Copy-PvsDeviceProperties

Copy-PvsDiskProperties

Copy-PvsServerProperties

Disable-PvsDeviceDiskLocator

Dismount-PvsDisk

Enable-PvsDeviceDiskLocator

Export-PvsAuditTrail

Export-PvsDisk

Export-PvsOemLicenses

Get-PvsADAccount

Get-PvsAuditActionParameter

Get-PvsAuditActionProperty

Get-PvsAuditActionSibling

Get-PvsAuditTrail

Get-PvsAuthGroup

Get-PvsAuthGroupUsage

Get-PvsCeipData

Get-PvsCisData

Get-PvsCollection

Get-PvsConnection

Get-PvsCreateDiskStatus

Get-PvsDevice

Get-PvsDeviceBootstrap

Get-PvsDeviceCount

Get-PvsDeviceDiskLocatorEnabled

Get-PvsDeviceDiskTempVersion

Get-PvsDeviceInfo

Get-PvsDevicePersonality

Get-PvsDeviceStatus

```
Get-PvsDirectory
Get-PvsDisk
Get-PvsDiskInfo
Get-PvsDiskInventory
Get-PvsDiskLocator
Get-PvsDiskLocatorCount
Get-PvsDiskLocatorLock
Get-PvsDiskUpdateDevice
Get-PvsDiskUpdateStatus
Get-PvsDiskVersion
Get-PvsExists
Get-PvsFarm
Get-PvsFarmView
Get-PvsGroup
Get-PvsLocalServer
Get-PvsMaintenanceVersionExists
Get-PvsMinimumLastAutoAddDeviceNumber
Get-PvsMountedDisk
Get-PvsMountedDriveLetter
Get-PvsNewDiskVersion
Get-PvsServer
Get-PvsServerBiosBootstrap
Get-PvsServerBootstrap
Get-PvsServerBootstrapName
Get-PvsServerCount
Get-PvsServerInfo
Get-PvsServerName
Get-PvsServerStatus
Get-PvsServerStore
Get-PvsServerStoreActiveDeviceCount
Get-PvsSite
Get-PvsSiteView
Get-PvsStore
Get-PvsStoreFreeSpace
Get-PvsStoreSharedOrServerPath
Get-PvsTask
Get-PvsTaskStatus
```

Get-PvsUndefinedDisk
Get-PvsUpdateTask
Get-PvsUploadCeip
Get-PvsVersion
Get-PvsVirtualHostingPool
Get-PvsXDSite
Grant-PvsAuthGroup
Import-PvsDevices
Import-PvsDisk
Import-PvsOemLicenses
Invoke-PvsActivateDeviceMAK
Invoke-PvsMarkDown
Invoke-PvsPromoteDiskVersion
Invoke-PvsRebalanceDevices
Invoke-PvsRevertDiskVersion
Merge-PvsDisk
Mount-PvsDisk
Move-PvsDeviceToCollection
Move-PvsServerToSite
New-PvsAuthGroup
New-PvsCeipData
New-PvsCisData
New-PvsCollection
New-PvsDevice
New-PvsDeviceWithPersonalvDisk
New-PvsDirectory
New-PvsDiskLocator
New-PvsDiskMaintenanceVersion
New-PvsDiskUpdateDevice
New-PvsFarmView
New-PvsServer
New-PvsSite
New-PvsSiteView
New-PvsStore
New-PvsUpdateTask
New-PvsVirtualHostingPool
Remove-PvsAuthGroup

Remove-PvsCollection
Remove-PvsDevice
Remove-PvsDeviceDiskCacheFile
Remove-PvsDeviceFromDomain
Remove-PvsDeviceFromView
Remove-PvsDirectory
Remove-PvsDiskFromUpdateTask
Remove-PvsDiskLocator
Remove-PvsDiskLocatorFromDevice
Remove-PvsDiskUpdateDevice
Remove-PvsDiskVersion
Remove-PvsFarmView
Remove-PvsServer
Remove-PvsServerStore
Remove-PvsSite
Remove-PvsSiteView
Remove-PvsStore
Remove-PvsUpdateTask
Remove-PvsVirtualHostingPool
Reset-PvsDatabase
Reset-PvsDeviceForDomain
Restart-PvsStreamService
Revoke-PvsAuthGroup
Set-PvsAuthGroup
Set-PvsCeipData
Set-PvsCisData
Set-PvsCollection
Set-PvsConnection
Set-PvsDevice
Set-PvsDeviceBootstrap
Set-PvsDevicePersonality
Set-PvsDisk
Set-PvsDiskLocator
Set-PvsDiskUpdateDevice
Set-PvsDiskVersion
Set-PvsFarm
Set-PvsFarmView

Set-PvsOverrideVersion

Set-PvsServer

Set-PvsServerBiosBootstrap

Set-PvsServerBootstrap

Set-PvsServerStore

Set-PvsSite

Set-PvsSiteView

Set-PvsStore

Set-PvsUpdateTask

Set-PvsVirtualHostingPool

Set-PvsXDSite

Start-PvsAutoUpdate

Start-PvsCreateDisk

Start-PvsDeviceBoot

Start-PvsDeviceDiskTempVersionMode

Start-PvsDeviceReboot

Start-PvsDeviceShutdown

Start-PvsDeviceUpdateBdm

Start-PvsDisplayMessage

Start-PvsReportBug

Start-PvsStreamService

Start-PvsUpdateTask

Stop-PvsCreateDisk

Stop-PvsDeviceDiskTempVersionMode

Stop-PvsStreamService

Stop-PvsTask

Stop-PvsUpdateTask

Test-PvsDirectory

Unlock-PvsAllDisk

Unlock-PvsDisk

Update-PvsInventory

Cmdlet by Object Type

Many

Clear-PvsConnection

Get-PvsAuditActionParameter

Get-PvsAuditActionProperty

Get-PvsAuditActionSibling

Get-PvsAuditTrail

Get-PvsConnection

Get-PvsExists

Reset-PvsDatabase

Set-PvsConnection

AuthGroup

Get-PvsAuthGroup

Get-PvsAuthGroupUsage

Grant-PvsAuthGroup

New-PvsAuthGroup

Remove-PvsAuthGroup

Revoke-PvsAuthGroup

Set-PvsAuthGroup

CeipData

Get-PvsCeipData

Get-PvsUploadCeip

New-PvsCeipData

Set-PvsCeipData

CisData

Get-PvsCisData

New-PvsCisData

Set-PvsCisData

Collection

Get-PvsCollection

Get-PvsMinimumLastAutoAddDeviceNumber

New-PvsCollection

Remove-PvsCollection

Set-PvsCollection

Device

Add-PvsDeviceToDomain

Add-PvsDeviceToView

Copy-PvsDeviceProperties

Disable-PvsDeviceDiskLocator

Enable-PvsDeviceDiskLocator

Export-PvsOemLicenses

Get-PvsDevice

Get-PvsDeviceBootstrap

Get-PvsDeviceCount

Get-PvsDeviceDiskLocatorEnabled

Get-PvsDeviceDiskTempVersion

Get-PvsDeviceInfo

Get-PvsDevicePersonality

Get-PvsDeviceStatus

Import-PvsDevices

Import-PvsOemLicenses

Invoke-PvsActivateDeviceMAK

Invoke-PvsMarkDown

Move-PvsDeviceToCollection

New-PvsDevice

New-PvsDeviceWithPersonalvDisk

Remove-PvsDevice

Remove-PvsDeviceDiskCacheFile

Remove-PvsDeviceFromDomain

Remove-PvsDeviceFromView

Reset-PvsDeviceForDomain

Set-PvsDevice

Set-PvsDeviceBootstrap

Set-PvsDevicePersonality

Start-PvsDeviceBoot

Start-PvsDeviceDiskTempVersionMode

Start-PvsDeviceReboot

Start-PvsDeviceShutdown

Start-PvsDeviceUpdateBdm

Start-PvsDisplayMessage

Stop-PvsDeviceDiskTempVersionMode

Disk

Add-PvsDiskToUpdateTask

Add-PvsDiskVersion

Copy-PvsDiskProperties

Dismount-PvsDisk

Export-PvsDisk
Get-PvsCreateDiskStatus
Get-PvsDisk
Get-PvsDiskInfo
Get-PvsDiskInventory
Get-PvsDiskUpdateDevice
Get-PvsDiskUpdateStatus
Get-PvsDiskVersion
Get-PvsMaintenanceVersionExists
Get-PvsMountedDisk
Get-PvsMountedDriveLetter
Get-PvsNewDiskVersion
Get-PvsUndefinedDisk
Get-PvsVirtualHostingPool
Import-PvsDisk
Invoke-PvsPromoteDiskVersion
Invoke-PvsRevertDiskVersion
Merge-PvsDisk
Mount-PvsDisk
New-PvsDiskMaintenanceVersion
New-PvsDiskUpdateDevice
New-PvsVirtualHostingPool
Remove-PvsDiskFromUpdateTask
Remove-PvsDiskUpdateDevice
Remove-PvsDiskVersion
Remove-PvsVirtualHostingPool
Set-PvsDisk
Set-PvsDiskUpdateDevice
Set-PvsDiskVersion
Set-PvsOverrideVersion
Set-PvsVirtualHostingPool
Start-PvsCreateDisk
Stop-PvsCreateDisk
Unlock-PvsAllDisk
Unlock-PvsDisk
Update-PvsInventory

DiskLocator

Add-PvsDiskLocatorToDevice

Get-PvsDiskLocator

Get-PvsDiskLocatorCount

Get-PvsDiskLocatorLock

New-PvsDiskLocator

Remove-PvsDiskLocator

Remove-PvsDiskLocatorFromDevice

Set-PvsDiskLocator

Farm

Export-PvsAuditTrail

Get-PvsFarm

Get-PvsXDSite

Set-PvsFarm

Set-PvsXDSite

FarmView

Get-PvsFarmView

New-PvsFarmView

Remove-PvsFarmView

Set-PvsFarmView

Server

Copy-PvsServerProperties

Get-PvsDirectory

Get-PvsServer

Get-PvsServerBiosBootstrap

Get-PvsServerBootstrap

Get-PvsServerBootstrapName

Get-PvsServerCount

Get-PvsServerInfo

Get-PvsServerName

Get-PvsServerStatus

Get-PvsServerStore

Get-PvsServerStoreActiveDeviceCount

Invoke-PvsRebalanceDevices

Move-PvsServerToSite

New-PvsDirectory

New-PvsServer

Remove-PvsDirectory

Remove-PvsServer

Remove-PvsServerStore

Restart-PvsStreamService

Set-PvsServer

Set-PvsServerBiosBootstrap

Set-PvsServerBootstrap

Set-PvsServerStore

Start-PvsAutoUpdate

Start-PvsReportBug

Start-PvsStreamService

Stop-PvsStreamService

Test-PvsDirectory

Site

Get-PvsSite

Get-PvsStoreSharedOrServerPath

New-PvsSite

Remove-PvsSite

Set-PvsSite

SiteView

Get-PvsSiteView

New-PvsSiteView

Remove-PvsSiteView

Set-PvsSiteView

Store

Get-PvsStore

Get-PvsStoreFreeSpace

New-PvsStore

Remove-PvsStore

Set-PvsStore

System

Get-PvsADAccount

Get-PvsGroup

Get-PvsLocalServer

Get-PvsVersion

Task

Clear-PvsTask

Get-PvsTask

Get-PvsTaskStatus

Stop-PvsTask

UpdateTask

Get-PvsUpdateTask

New-PvsUpdateTask

Remove-PvsUpdateTask

Set-PvsUpdateTask

Start-PvsUpdateTask

Stop-PvsUpdateTask

Error codes

For the Citrix.PVS.SnapIn, if an error occurs, a PvsException will be in the Exception member of the $error.

The members of a PvsException are:

InnerException: The exception that occured. This exception maybe an EAException or other standard Exception.

ToString(): Has the formatted full Message of the InnerException.

If the InnerException GetType().Name equals "EAException", then The members of it are:

returnCode: The number, as shown below in the Error codes. The name of the error, for example "NotImplemented", is not included in the EAException.

Message: The message, as shown below in the Error codes. The [v1], [v2], [v3], [v4], and [v5] will be replaced with values as required.

Details: Has the Details for the EAException if there are any. OtherException, ManagementInterfaceError and PvsStatusException will have Details.

ToString(): Has the Message as shown below in the Error codes. If there is Details, it will be returned or included, and if partialReturn, they will be included.

partialReturn: Might have a list of EAException objects if any of the items processed during the command had any issues.

Severity: Can have the values Critical, Error, Warning or Duplicate.

Source: Has the value that is displayed in the Console as a Title or Type for the error.

0 Success: The command succeeded.

1 NotImplemented: The [v1] feature has not been implemented.

2 InvalidCommand: The [v1] command does not exist.

3 InvalidField: The [v1] field does not exist.

4 InvalidFieldFormat: The [v1] field is not formatted properly, the correct format is [v2].

5 InvalidParameter: The [v1] parameter is not valid.

6 InvalidParameterFormat: The [v1] parameter is not formatted property, the correct format is [v2].

7 ReadOnlyField: Unable to change the [v1] field because it is read-only.

8 RequiredFieldMissing: The required [v1] field is missing.

9 RequiredFieldsMissing: The required [v1] or [v2] field is missing.

10 RequiredParameterMissing: The required [v1] parameter is missing.

11 RequiredParametersMissing: The required [v1] or [v2] parameter is missing.

12 InternalIdAndNameFieldsMustBeDefined: An internal error occurred. The [v1] field is not the next FieldSettings object after the ID.

13 NoFarmAccess: The domain/user does not have access to the Farm.

14 InvalidForeignKeyValue: The [v1] field with value [v2] is an invalid foreign key.

15 SetupError: The system was not configured correctly.

16 Executing: The [v1] command can only be called one at a time. Wait for the command to finish before running again.

17 NoDiskMapped: A vDisk has not yet been mapped.

18 DatabaseError: A database error occurred.

19 DuplicateKey: To avoid creating a duplicate key, the Add or Set command was cancelled.

20 DatabaseErrorMissed: An internal error occurred. An uncaught database error occurred.

21 AddCommandFailed: No objects were added during the last 'Add' command.

22 InsufficientPrivileges: Access denied. The appropriate privileges are not assigned to perform this task.

23 ZeroObjectsAffected: No object was added, updated, or deleted in the last operation.

24 OtherException: An unexpected MAPI error occurred.

25 InvalidFieldLength: The [v1] field value is too long, maximum length is [v2].

26 InvalidFieldValueMinMax: The [v1] field value is invalid, the minimum is [v2] and the maximum is [v3].

27 InvalidFieldValue: The [v1] field can only have values [v2] or [v3].

28 TooManyParameters: More parameters were specified than permitted.

29 TooFewParameters: Not enough identifying parameters specified.

30 FollowingParametersMissing: To use the [v1] parameter, [v2] or [v3] must also be used.

31 InconsistentData: The action is canceled because the Store directory date/times does not match. Update the Store directories to match.

32 DatabaseOpenFailed: Unable to contact the database server. Ensure Provisioning Services is configured correctly.

33 DatabaseVersionWrong: The wrong database version is being used. Found version number: [v1] Expected version number: [v2]

34 DatabaseVersionNotFound: The database version number does not exist or was not found. Ensure Provisioning Services is configured correctly.

35 SomeRequiredParametersMissing: Required parameters are missing.

36 PartialError: The following items failed:

item1 Error message...

item2 Error message...

37 NoItemsToProcess: There are no items to process.

38 NoDefaultCollectionDefined: Unable to add a Device until a default Collection is set.

39 NoDefaultSiteDefined: A default Site is not set, no Devices can be added.

40 InvalidCollection: The specified Collection does not exist.

41 InvalidAuthGroup: The specified AuthGroup does not exist.

42 InvalidGroup: The specified Group does not exist.

43 InvalidDevice: The specified Device does not exist.

44 InvalidDiskLocator: The specified vDisk does not exist.

45 InvalidServer: The specified Server does not exist.

46 InvalidServerSite: Server specified is not in the Site specified.

47 InvalidStoreSite: Store specified is not for the Site specified.

48 InvalidSiteView: The specified Site View does not exist.

49 InvalidSite: The specified Site does not exist.

50 InvalidDeviceDiskLocator: The specified Device or vDisk does not exist.

51 InvalidDeviceImport: Import failed because the file must have Device Name, Mac Address, Site Name, and Collection Name, and they must be tab or comma-delimited.

52 InvalidServerFrom: The Server to copy [v1]=[v2] was not found.

53 InvalidServerTo: No Server to copy to ([v1]=[v2]) was found.

54 InvalidDeviceFrom: The Device to copy [v1]=[v2] was not found.

55 InvalidDeviceTo: No Devices to copy to are found.

56 InvalidDiskFrom: The vDisk to copy [v1]=[v2] was not found.

57 InvalidDiskTo: No vDisk to copy to ([v1]=[v2]) was found.

58 InvalidDiskPath: The path '[v1]' to the vDisk file is not found.

59 VDiskFileNotFound: [v1]: vDisk file was not found.

60 InvalidDiskServer: There is no Server that can serve the vDisk [v1] or the Store to which this vDisk belongs. Verify that one or more Servers belonging to the Store are online and that there is sufficient free space for the operation you are attempting.

61 InvalidDiskForServer: Server [v1] cannot access all versions of vDisk [v2], the vDisk was updated on at least one other Server.

62 SameSiteRequired: Objects within the same Site must be selected.

63 TooFewFields: Not enough fields for a record.

64 ADerrorDC: Unable to connect to the Domain Controller (if any) or the default rootDSE. Error code: [v1], message: [v2], provider: [v3].

65 ADerrorOU: Unable to get the Organizational Unit setting (if any). Error code: [v1], message: [v2], provider: [v3].

66 ADerrorDefaultContainer: Unable to get the default computer accounts container (default location is Active Directory root> Computers). Error code: [v1], message: [v2], provider: [v3].

67 ADerrorCreate: Unable to create the computer account in Active Directory. Ensure the account does not already exist and that the appropriate permissions are available to perform this task. Error code: [v1], message: [v2], provider: [v3].

68 ADerrorNewAccount: Unable to get the newly created Active Directory computer account. Error code: [v1], message: [v2], provider: [v3].

69 ADerrorSam: Unable to set the Active Directory samAccountName property. Ensure the appropriate permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

70 ADerrorUserAccount: Unable to set the Active Directory userAccountControl property. Ensure the appropriate permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

71 ADerrorSave: Unable to save Active Directory change. Ensure the appropriate permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

72 ADerrorSetPassword: Unable to set a new password for this user account. Ensure the appropriate permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

73 ADerrorAddTrustee: Unable to add trustee (if any). Error code: [v1], message: [v2], provider: [v3].

74 ADerrorEnableAccount: Unable to enable the Active Directory account. Error code: [v1], message: [v3], provider: [v2].

75 ADerrorAlreadyExists: The computer name is already in use. Error code: [v1], message: [v3], provider: [v2]. Select a unique name for this machine.

76 ADerrorGeneral: A general Active Directory error occurred. Error code: [v1], message: [v2], provider: [v3].

77 ADerrorDirectorySearch: Unable to find Active Directory items meeting the search criteria entered. Error code: [v1], message: [v2], provider: [v3].

78 ADerrorSearchComputerAccount: Unable to perform the computer accounts search. Error code: [v1], message: [v2], provider: [v3].

79 ADerrorComputerAccountNotFound: Specified computer account not found. Error code: [v1], message: [v2], provider: [v3].

80 ADerrorComputerAccountHold: This computer account is currently unavailable. Ensure that Active Directory is running properly. Error code: [v1], message: [v2], provider: [v3].

81 ADerrorComputerAccountMove: Failed to move the computer account to the target organizational unit set (also returned if caller lacks permission). Error code: [v1], message: [v2], provider: [v3].

82 ADerrorDelete: Unable to delete this computer account. Ensure the appropriate permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

83 ADerrorPasswordGeneration: Unable to generate this password. Ensure the appropriate permissions exist to perform this task.

84 MapDiskNoDriver: Unable to map vDisk because a driver was not found.

85 MapDiskDeniedByServer: Unable to map the vDisk. Mapping was denied by the Server.

86 MapDiskLocalAccessDenied: Unable to map the vDisk. Denied local access.

87 MapDiskMiniportError: Unable to map vDisk because of a Miniport error.

88 UnmapDiskFailed: Failed to unmap a vDisk.

89 DuplicateDisk: The vDisk [v1] already exists on [v2] at [v3].

90 DuplicateDiskLocator: A DiskLocator: [v1] already exists on Site: [v2].

91 DiskCreationInProgress: The vDisk [v1] is being created on [v2] at [v3].

92 InvalidServerStore: A database integrity error occurred. The Server is not set to deliver vDisks from the Store, but should be.

93 InvalidStore: The specified Store does not exist.

94 InvalidFarmView: Farm View specified does not exist.

95 InvalidStorePath: Store path is empty.

96 ManagementInterfaceError:

Management Interface: Undefined error.

Management Interface: Database interface is inaccessible.

Management Interface: Database interface library is inaccessible.

Management Interface: The database access library is a version incompatible with the Management Server.

Management Interface: Database interface library is invalid.

Management Interface: Database interface could not be created.

Management Interface: Database could not be opened.

Management Interface: Database is in use.

Management Interface: Database error occurred.

Management Interface: Not implemented.

Management Interface: Registry entry was not found.

Management Interface: Request was not created.

Management Interface: Operating System error occurred.

Management Interface: vDisk error.

Management Interface: vDisk header is incomplete.

Management Interface: vDisk footer is incomplete.

Management Interface: vDisk boot record is incomplete.

Management Interface: vDisk boot sector is incomplete.

Management Interface: vDisk size is below the minimum.

Management Interface: vDisk size is above the maximum.

Management Interface: vDisk boot record template is inaccessible.

Management Interface: vDisk boot sector template is inaccessible.

Management Interface: vDisk lock was not found.

Management Interface: vDisk has exclusive lock.

Management Interface: vDisk has shared lock.

Management Interface: vDisk lock error.

Management Interface: vDisk format is incompatible.

Management Interface: vDisk prefooter is incomplete.

Management Interface: vDisk creation is in progress.

Management Interface: vDisk creation information was not found.

Management Interface: vDisk creation cancellation was requested.

Management Interface: vDisk file was not found.

Management Interface: vDisk file path was not found.

Management Interface: vDisk file access was denied.

Management Interface: Cancelled.

Management Interface: Registry key for the product is inaccessible.

Management Interface: Registry key for the installation folder is inaccessible.

Management Interface: Registry key for the management interface is inaccessible.

Management Interface: Registry key for the database path is inaccessible.

Management Interface: Registry key for the management interface IP address is inaccessible.

Management Interface: Buffer size is too small.

Management Interface: Buffer size is too large.

Management Interface: Unknown error.

Management Interface: Remote Server failed to relay a request.

Management Interface: Remote Server is not servicing the Device.

Management Interface: Remote Server or Device refused the request.

Management Interface: Local Server failed to complete a request to a Server or Device.

Management Interface: Local Server failed to complete a request to a Server.

Management Interface: Remote requests were disabled because of an initialization error.

Management Interface: Remote request failed.

Management Interface: Remote request timed out.

Management Interface: Remote request result was not found.

Management Interface: Remote request receiver failed to initialize.

Management Interface: Management command failed for all objects.

Management Interface: Failed to get the preshared key in secure version.

Management Interface: VHD Error.

Management Interface: vDisk properties were lost.

Management Interface: Insufficient Memory.

Management Interface: The network path was not found.

Management Interface: The network name cannot be found.

Management Interface: File already exists.

Management Interface: The geometry of the vDisk is not accessible.

Management Interface: Unable to create the vDisk because the store media is read-only.

Management Interface: vDisk file is being used by another process.

97 ServerTimeout: Server did not respond to a request in time.

98 NotFound: [v1] not found.

99 AccountRetrieve: Account information for user [v1] was not found.

100 ActiveDevice: The task cannot be performed on active Devices. Shut down the Devices before attempting to perform the task.

101 ActiveDiskLocator: The task cannot be performed on active vDisks. Shut down the Devices that are using the vDisks before attempting to perform the task.

102 AssignedDiskLocator: Unable to delete a vDisk that is currently assigned to a Device. Unassign all Devices, then delete the vDisk.

103 ActiveServer: The task cannot be performed on active Servers. Shut down the Servers before attempting to perform the task.

104 NotEnoughFreeDiskSpace: There is not enough free disk space to create the vDisk.

105 InvalidDiskName: The vDisk name has one or more invalid characters. The invalid characters are < > | " \ / : * ?.

106 CannotDeleteLastAuthGroup: Deleting the last Authorization Group causes the system to be inoperable.

107 CannotDeleteUsedAuthGroup: An Authorization Group that is currently in use cannot be deleted.

108 ServerStartFailed: The Server did not start successfully. Ensure the appropriate permissions exist for the service account.

109 ServerStopFailed: The Server did not stop successfully.

110 LockOwnerNotFound: The Device that owns the lock was not found, the vDisk was not unlocked.

111 PossiblySharedVDisk: Unable to delete File [v1]. It is possible that the file is being referenced in other Sites or Stores.

112 StorePathInaccessible: The Store path [v1] is inaccessible.

113 InvalidAction: The [v1] action does not exist.

114 InvalidObjectType: The [v1] objectType does not exist.

115 TooManyRecords: The amount of data returned using Get is too large. Use GetFirst and GetNext instead of Get.

116 InvalidUserGroup: The specified UserGroup does not exist.

117 InvalidAuditAction: The specified AuditAction does not exist.

118 LoginFailed: The database login failed. Ensure the appropriate permissions exist to access the database.

119 DatabaseConnectionError: Unable to connect to the database. Restore the connection in order to manage the farm.

120 CreateTriggersParsing: Unable to parse the database script 'CreateTriggers' at: [v1]

121 CreateStoredProcParsing: Unable to parse the database script 'CreateStoredProcedures' at: [v1]

122 MediaIsReadOnly: Management Interface: Unable to create the vDisk because the store media is read-only.

123 ConnectedDeviceForVirtualHostingPool: Unable to delete this VM from a machine catalog because it is connected to a Delivery Group.

124 ADerrorDN: Unable to get the distinguishedName property. Ensure the appropriate Active Directory permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

125 ADerrorGetSecDes: Unable to get the Active Directory Security Descriptor property. Error code: [v1], message: [v2], provider: [v3].

126 ADerrorSetSecDes: Unable to set the Active Directory Security Descriptor property. Ensure the appropriate permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

127 ADerrorDNSHostName: Unable to set the DNS Host Name property (dNSHostName). Ensure the appropriate permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

128 ADerrorDisplayName: Unable to set the displayName property. Error code: [v1], message: [v2], provider: [v3].

129 ADerrorBind: This device was unable to bind to the Domain Controller. Ensure the Domain Controller is running. Error code: [v1], message: [v2], provider: [v3].

130 ADerrorGetSPN: Unable to get an Active Directory Service Principal Name. Error code: [v1], message: [v2], provider: [v3].

131 ADerrorWriteSPN: Unable to write the Active Directory Service Principal Name. Error code: [v1], message: [v2], provider: [v3]

132 ADerrorSearch: Unable to perform the requested Search. Error code: [v1], message: [v2], provider: [v3].

133 ADerrorMoveToOU: Unable to move the Active Directory account to the requested Organizational Unit. Ensure the appropriate permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

134 ADerrorDeleteAccount: Unable to delete this computer account. Ensure the appropriate permissions exist to delete accounts. Error code: [v1], message: [v2], provider: [v3].

135 ADerrorBadParameters: Incorrect parameters sent to Provisioning Services from Studio. Error code: [v1], message: [v2], provider: [v3].

136 VolumeInUse: The volume is being used.

137 VolumeAccessDenied: Volume access is denied.

138 VolumeUnknownVolume: An unknown volume was specified.

139 VolumeGeneralError: An error occured when executing a volume command.

140 MaintenanceServerError: Action cannot be performed, [v1] is a maintenance server for [v2].

141 NotManagedStore: The action cannot be performed because the store is not managed.

142 PathNotExist: The path does not exist on the given Server.

143 PathNoCreatePermission: The path does not have the appropriate create permissions.

144 PathNoReadPermission: The path does not have the appropriate read permissions.

145 PathNoWritePermission: The path does not have the appropriate write permissions.

146 PathNoDeletePermission: The path does not have the appropriate delete permissions.

147 IPCProtocolError: An internal error occurred. A field is missing from the process communication protocol data.

148 InvalidStoreServer: No active Server can serve the Store [v1].

149 ConstraintCheck: A database constraint caused an Add or Update to be stopped.

150 VamtNotFound: The Volume Activation Management Tool cannot be found.

151 ADerrorCannotGetObjectSID: Cannot return objectSID. Error code: [v1], message: [v2], provider: [v3].

152 ADerrorCannotDisableAccount: Cannot disable the Active Directory account at this time. Ensure that all account users are logged off before attempting to disable the account. Error code: [v1], message: [v2], provider: [v3].

153 ADerrorFailedToChangePassword: Unable to reset the machine account password. Ensure the appropriate permissions exist to perform this Active Directory task. Error code: [v1], message: [v2], provider: [v3].

154 ADerrorFailedToCopyDCName: Unable to copy the Domain Controller name. Error code: [v1], message: [v2], provider: [v3].

155 ADerrorDCNameIsTooLong: The Domain Controller name entered exceeds the maximum character length of [v4]. Error code: [v1], message: [v2], provider: [v3].

156 SiteMakUserPassword: The Site's makUser and makPassword fields must have values.

157 VamtError: See the log for additional error details.

158 InactiveDevice: Device specified is not active.

159 DiskIsInPrivateMode: This task cannot be performed because the vDisk is in private image mode.

160 AlreadyInChangeMode: Unable to complete this operation, vDisk is already in Maintenance, Merge, or Test mode.

161 CannotCreateMaintenanceDisk: Cannot create maintenance vDisk.

162 CannotEnterMaintenanceMode: To place a vDisk in Maintenance Mode requires using a Server. No Server is available at this time.

163 NotInMaintenanceMode: Unable to perform this action because the vDisk is not in Maintenance Mode.

164 NoVersionForMaintenanceMode: Unable to place this vDisk in Maintenance Mode because the highest version is not found.

165 NoVersionFound: Unable to perform this action because a version record was not found in the database.

166 Obsolete: The [v1] feature is obsolete.

167 DatabaseWarning: A database warning occurred.

168 DatabaseSQL: A database SQL error occurred.

169 DatabaseResource: A database resource error occurred.

170 InvalidUpdateTask: The specified UpdateTask does not exist.

171 InvalidVirtualHostingPool: The specified VirtualHostingPool does not exist.

172 RemoteCommand: An exception occurred executing a command on a remote Server.

173 IpcNotConfigured: An internal error occurred. The process communication interface must be configured before executing remote commands.

174 DiskAlreadSetForUpdate: The vDisk is already set for Update with Device [v1] in Site [v2].

175 InvalidDiskVersion: The vDisk Version specified is not valid.

176 HostResolution: Could not resolve the host name for [v1].

177 InProcess: The remote task is taking longer than expected. TaskId: [v1]

178 DateMustBeInFuture: The [v1] must be in the future.

179 InvalidRemoteReturn: The remote command did not return valid data.

180 InvalidParameterValueMinMax: The [v1] parameter value is invalid, the minimum is [v2] and the maximum is [v3].

181 InvalidParameterNotNumeric: The [v1] parameter value is invalid, it is not numeric.

182 InvalidParameterNotGuid: The [v1] parameter value is invalid, it is not a GUID.

183 PassThroughMessage: [v1]

184 DiskUpdateNotEnabled: The Automatic vDisk Update option must be enabled and the vDisk Update Server must be defined. Set these in the Site properties.

185 PvsStatusException:

Windows API error occurred, number 0xE000FFFF.

SQL error occurred, number 0xE001FFFF.

Manager error occurred. Error number 0xE002FFFF.

StreamProcess error occurred. Error number 0xE003FFFF.

Stream Database error occurred. Error number 0xE004FFFF.

Management error occurred. Error number 0xE005FFFF.

Shutdown in progress; request ignored. Error number 0xE0050001.

CreateDiffDisk: Malformed packet; missing one or more arguments. Error number 0xE0050002.

DeleteDiffDisk: Malformed file name; cannot parse directory and name. Error number 0xE0050003.

DeleteDiffDisk: Malformed packet; missing one or more arguments. Error number 0xE0050004.

IPC: Failed to read mtGetLocks parameters. Error number 0xE0050005.

IPC: Failed to read mtGetLockStatus parameters. Error number 0xE0050006.

IPC: Failed to read mtLock parameters. Error number 0xE0050007.

IPC: Failed to read mtUnlock parameters. Error number 0xE0050008.

MergeDisk event: Malformed packet; unknown message type. Error number 0xE0050009.

MergeDisk event: Unknown target request ID. Error number 0xE005000A.

MergeDisk event: Malformed packet; missing one or more arguments. Error number 0xE005000B.

MergeDisk: Malformed packet; missing one or more arguments. Error number 0xE005000C.

ValidateDisk: Malformed packet; missing one or more arguments. Error number 0xE005000D.

VHD Library error occurred. Error number 0xE006FFFF.

VHD Library: Not implemented. Error number 0xE0060001.

VHD Library: Handle pointer is invalid. Error number 0xE0060002.

VHD Library: Length of the path exceeds the limit of the file system. Error number 0xE0060003.

VHD Library: Name is empty. Error number 0xE0060004.

VHD Library: Length of the name exceeds the limit of the file system. Error number 0xE0060005.

VHD Library: Size of a parameter was too big. Error number 0xE0060006.

VHD Library: Size of a parameter was too small. Error number 0xE0060007.

VHD Library: The media is write protected. Error number 0xE0060008.

VHD Library: Type is invalid. Error number 0xE0060009.

VHD Library: Footer is incomplete. Error number 0xE006000A.

VHD Library: Failed to read or write the entire VHD Header. Error number 0xE006000B.

VHD Library: Failed to read or write the entire VHD Block Allocation Table. Error number 0xE006000C.

VHD Library: Failed to read or write all of the VHD properties. Error number 0xE006000D.

VHD Library: VHD footer is corrupt. Error number 0xE006000E.

VHD Library: VHD header is corrupt. Error number 0xE006000F.

VHD Library: Failed to read or write the VHD objects. Error number 0xE0060010.

VHD Library: Destination string is too small. Error number 0xE0060011.

VHD Library: Destination string pointer is NULL. Error number 0xE0060012.

VHD Library: Source string pointer is NULL. Error number 0xE0060013.

VHD Library: Offset is before the beginning of the VHD data area. Error number 0xE0060014.

VHD Library: Offset is after the end of the VHD data area. Error number 0xE0060015.

VHD Library: Failed to allocate memory because it was unavailable. Error number 0xE0060016.

VHD Library: Caller cancelled the last create request. Error number 0xE0060017.

VHD Library: Failed to read or write all of the data as requested. Error number 0xE0060018.

VHD Library: Failed to create a Universal Unique Identification for a VHD. Error number 0xE0060019.

VHD Library: Failed to find the VHD properties. Error number 0xE006001A.

VHD Library: Failed to read or write the entire sector bitmap within a block. Error number 0xE006001B.

VHD Library: Failed to read or write the entire block. Error number 0xE006001C.

VHD Library: Failed to open the file that represents the VHD. Error number 0xE006001D.

VHD Library: Requested number of bytes exceeds the remainder of bytes in a block. Error number 0xE006001E.

VHD Library: Accessed past end of the VHD file. Error number 0xE006001F.

VHD Library: Differencing VHD Unique ID (UUID) differs to parent VHD Unique ID. Error number 0xE0060020.

VHD Library: Differencing VHD timestamp differs to parent VHD last modified time. Error number 0xE0060021.

VHD Library: Failed to read or write the entire VHD Block Allocation Table Map. Error number 0xE0060022.

IPC error occurred. Error number 0xE007FFFF.

There was an unknown transmission error. Error number 0xE0070001.

No response received for successful send. Error number 0xA0070002.

Message processor timed out. Error number 0xE0070003.

Retry limit exhausted. Error number 0xE0070004.

Message recipient task is not active. Error number 0xE0070005.

Socket send/recv cannot be retried. Error number 0xE0070006.

Port shutdown due to connection opens exhausted. Error number 0xE0070007.

Port shutdown due to flood of junk packets. Error number 0xE0070008.

Port shutdown due to receive retries exhausted. Error number 0xE0070009.

Transport does not support fragmentation. Error number 0xE007000A.

One or more packet fragments are missing. Error number 0xE007000B.

Error sending message. Error number 0xE0070100.

Message acknowledgement timeout. Error number 0xA0070101.

Command timeout. Error number 0xE0070102.

Not implemented. Error number 0xE0070103.

Error verifying message port number, must be >= 0 and <= 65535. Error number 0xE0070104.

Command initialization failed. Error number 0xE0070105.

Start of IPC failed. Error number 0xE0070106.

Stop of IPC failed. Error number 0xE0070107.

Memory allocation failure. Error number 0xE0070108.

Internal error, failure to wait long enough for a communication response to be received. Error number 0xE0070109.

Disk Update error occurred. Error number 0xE008FFFF.

Inventory error occurred. Error number 0xE009FFFF.

Inventory Table: Failed to start thread. Error number 0xE0090001.

Inventory Table: Invalid Entry. Error number 0xE0090002.

Inventory Table: Failed to initialize inventory. Error number 0xE0090003.

Shutdown in progress; request ignored. Error number 0xE0090004.

Get Disk Inventory: Parameters bad. Error number 0xE0090033.

Populate database: Failed offline. Error number 0xE0090065.

Populate database: Server get by name failed. Error number 0xE0090066.

Populate database: Uninitialized. Error number 0xE0090067.

Populate database: Get host name failed. Error number 0xE0090068.

Populate database: Char conversion failed. Error number 0xE0090069.

Populate database: Initialization failed. Error number 0xE009006A.

Populate database: Database open failed. Error number 0xE009006B.

Populate database: Get all disk locators failed. Error number 0xE009006C.

Inventory Table: Not yet implemented. Error number 0xE009006D.

Notifier error occurred. Error number 0xE00AFFFF.

MAPI error occurred. Error number 0xE00BFFFF.

186 TaskCancelled: Task [v1] is cancelled and is not running.

187 TaskCompleted: Task [v1] has been completed and is not running.

188 TaskInProgress: Task [v1] is running and cannot be processed.

189 InvalidTask: The specified Task does not exist.

190 InventoryServerCannotContactDatabase: The Inventory Service cannot contact the database.

191 ServerOffline: The Server is offline.

192 ServerStateUnknown: The Server state is unknown.

193 HighestVersionIsPending: Could not complete this action because the highest vDisk version is still pending. The scheduled date for the version has not occurred yet.

194 MergeInvalidWithCurrentVersions: Merge is not valid with the current versions that exist.

195 DiskInventoryError: vDisk versions are not up to date on all Servers that access this vDisk. Update all Servers with the latest versions of the vDisk files.

196 VDiskFileNotFoundWarning: [v1]: vDisk file was not found because it was deleted.

197 CannotAssignActiveServer: Stop the Server before attempting to assign the Server to a different site.

198 CannotAssignServerWithActiveDevice: Before attempting to assign the Server to a different site, shut down Devices connecting to the Server, then shut down the Server.

199 MappedDiskLocator: The vDisk is mapped and cannot be changed.

200 InvalidTemplateDevice: The Template Device must be a Production Device that does not have a Personal vDisk.

201 DeviceWithPersonalVDiskInvalid: Unable to process a Device that uses a personal vDisk.

202 CreatingDisk: Server is creating a vDisk so change cannot be done.

203 AssignedDiskLocatorToDeviceWithPersonalvDisk: Unable to delete a vDisk if the vDisk is currently assigned to a Device that uses a Personal vDisk. Unassign the Device, then delete the vDisk.

204 InvalidMacAddress: The MAC address for this VM is invalid. Configure the VM with a valid MAC address.

205 CannotGetMacFromHypervisor: The hypervisor did not return the MAC address for this VM: [v1]

206 Win32SystemException: A system error occurred.

207 RemoteManagementIpCannotBeResolved: Unable to resolve the management IP for Server [v1].

208 LocalManagementIpNotSet: The management IP for local server [v1] is not set in registry IPC\IPv4Address.

209 PerformVolumeMaintenanceTaskPermissions: Ensure the Service Account user has the appropriate 'Perform volume maintenance task' permissions.

210 CannotLoginToVirtualHostingPool: Unable to log on to the virtual hosting pool [v1]. Ensure that the hypervisior server is running properly.

211 VirtualHostingPoolNotSetForDevice: The virtualHostingPoolId for device [v1] with bdmBoot must be set.

212 ActiveBdmBootDeviceCannotProcess: The Boot Device Manager [v1] did not process successfully.

213 CannotMovePvdDeviceToAnotherSite: Personal vDisk Devices cannot be moved to another site.

214 XenDesktopSiteInvalid: XenDesktop Site for Devices is not valid, the XenDesktop Site is: [v1]

215 XenDesktopServiceListOutOfDate: XenDesktop Site [v1] is not reachable, check that the Citrix PVS Soap Server service user has XenDesktop permissions and network connectivity.

216 NoXenDesktopServiceForPersonalVDiskCapability: No XenDesktop service found for Personal vDisk capability.

217 InsufficientPermissionsToPreparePersonalVDisks: The user account for the Citrix PVS Soap Server has insufficient permissions to prepare Personal vDisks.

218 NotEnoughFreeDiskSpaceForManifest: There is not enough free disk space to create the manifest.

219 OperationCannotBeDoneOnlyPvdDevicesAssigned: Operation cannot be done, only Personal vDisk Devices are assigned.

220 DiskFormatCannotBeSetToVHD: The format cannot be set to VHD since no VHD vDisk file is found in the path, [v1], for Server, [v2].

221 DiskFormatCannotBeSetToVHDX: The format cannot be set to VHDX since no VHDX vDisk file is found in the path, [v1], for Server, [v2].

222 TemporaryVersionIsSet: This task cannot be performed because a temporary version is set.

223 DiskIsUsingPersistentCacheOnServer: A temporary version cannot be used for a vDisk that is using persistent cache on server.

224 UploadAlreadyInProgress: An upload is already in progress by Server [v1].

225 FieldMustBeNull: Field [v1] must be null.

226 DuplicateData: Record already exists in [v1] table for Farm.

227 CisUploadTokenGenerateError: Error generating upload token for My Citrix username [v1] ([v2]).

228 InvalidCredentials: The username or password is incorrect.

229 NoWriteAccessToFolders: No write access to folders [v1] or [v2].

230 ReportCreationError: Error creating problem report: [v1].

231 PvsProxyNotSupported: PVS Proxy not supported on this host

232 CannotCreateRegKey: Cannot create Registry key [v1]

4100 ADerrorUnexpectedError: An unexpected Active Directory related error occured. Ensure the appropriate permissions exist to perform this task. Error code: [v1], message: [v2], provider: [v3].

Objects, in the Citrix.PVS.SnapIn Namespace

PvsADAccount

Read-Only Fields

string Domain: Domain the account is a member of.

string DomainController: The name of the DC used to create the host's computer account.

string Name: Name of the Device for the account.

string Sid: The value of the objectSID AD attribute of the same name for the Device's computer account.

PvsAuditAction

Read-Only Fields

Guid Guid or AuditActionId: GUID of the action.

Guid ObjectId: GUID of the object of the action.

string ObjectName: Name of the object of the action. Max Length=1000

string Path: Path of the object of the action. An example is Site\Collection for a Device. Default="" Max Length=101

Guid SiteId: GUID of the Site for the object of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

Guid SubId: GUID of the Collection or Store of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

uint Type: Type of object that action was performed on. Values are: 1 (AuthGroup), 2 (Collection), 3 (Device), 4 (Disk), 5 (DiskLocator), 6 (Farm), 7 (FarmView), 8 (Server), 9 (Site), 10 (SiteView), 11 (Store), 12 (System), and 13 (UserGroup)

PvsAuditActionParameter

Read-Only Fields

Guid AuditActionId: GUID of the Audit Action used for Get and Set.

string Name or AuditParameterName: Name of the parameter. Max Length=50

string Value: Value of the parameter. Max Length=1000

PvsAuditActionProperty

Read-Only Fields

Guid AuditActionId: GUID of the Audit Action used for Get and Set.

string Name or AuditPropertyName: Name of the property. Max Length=50

string NewValue: New value of the Property. Default="" Max Length=1000

string OldValue: Previous value of the Property. Default="" Max Length=1000

PvsAuditTrail

Read-Only Fields

uint Action: Name of the action taken. This is a number that is converted to a string for display. Values are: 1 (AddAuthGroup), 2 (AddCollection), 3 (AddDevice), 4 (AddDiskLocator), 5 (AddFarmView), 6 (AddServer), 7 (AddSite), 8 (AddSiteView), 9 (AddStore), 10 (AddUserGroup), 11 (AddVirtualHostingPool), 12 (AddUpdateTask), 13 (AddDiskUpdateDevice), 1001 (DeleteAuthGroup), 1002 (DeleteCollection), 1003 (DeleteDevice), 1004 (DeleteDeviceDiskCacheFile), 1005 (DeleteDiskLocator), 1006 (DeleteFarmView), 1007 (DeleteServer), 1008 (DeleteServerStore), 1009 (DeleteSite), 1010 (DeleteSiteView), 1011 (DeleteStore), 1012 (DeleteUserGroup), 1013 (DeleteVirtualHostingPool), 1014 (DeleteUpdateTask), 1015 (DeleteDiskUpdateDevice), 1016 (DeleteDiskVersion), 2001 (RunAddDeviceToDomain), 2002 (RunApplyAutoUpdate), 2003 (RunApplyIncrementalUpdate), 2004 (RunArchiveAuditTrail), 2005 (RunAssignAuthGroup), 2006 (RunAssignDevice), 2007 (RunAssignDiskLocator), 2008 (RunAssignServer), 2009 (RunWithReturnBoot), 2010 (RunCopyPasteDevice), 2011 (RunCopyPasteDisk), 2012 (RunCopyPasteServer), 2013 (RunCreateDirectory), 2014 (RunCreateDiskCancel), 2015 (RunDisableCollection), 2016 (RunDisableDevice), 2017 (RunDisableDeviceDiskLocator), 2018 (RunDisableDiskLocator), 2019 (RunDisableUserGroup), 2020 (RunDisableUserGroupDiskLocator), 2021 (RunWithReturnDisplayMessage), 2022 (RunEnableCollection), 2023 (RunEnableDevice), 2024 (RunEnableDeviceDiskLocator), 2025 (RunEnableDiskLocator), 2026 (RunEnableUserGroup), 2027 (RunEnableUserGroupDiskLocator), 2028 (RunExportOemLicenses), 2029 (RunImportDatabase), 2030 (RunImportDevices), 2031 (RunImportOemLicenses), 2032 (RunMarkDown), 2033 (RunWithReturnReboot), 2034 (RunRemoveAuthGroup), 2035 (RunRemoveDevice), 2036 (RunRemoveDeviceFromDomain), 2037 (RunRemoveDirectory), 2038 (RunRemoveDiskLocator), 2039 (RunResetDeviceForDomain), 2040

(RunResetDatabaseConnection), 2041 (RunRestartStreamingService), 2042 (RunWithReturnShutdown), 2043 (RunStartStreamingService), 2044 (RunStopStreamingService), 2045 (RunUnlockAllDisk), 2046 (RunUnlockDisk), 2047 (RunServerStoreVolumeAccess), 2048 (RunServerStoreVolumeMode), 2049 (RunMergeDisk), 2050 (RunRevertDiskVersion), 2051 (RunPromoteDiskVersion), 2052 (RunCancelDiskMaintenance), 2053 (RunActivateDevice), 2054 (RunAddDiskVersion), 2055 (RunExportDisk), 2056 (RunAssignDisk), 2057 (RunRemoveDisk), 2058 (RunDiskUpdateStart), 2059 (RunDiskUpdateCancel), 2060 (RunSetOverrideVersion), 2061 (RunCancelTask), 2062 (RunClearTask), 2063 (RunForceInventory), 2064 RunUpdateBDM, 2065 (RunStartDeviceDiskTempVersionMode), 2066 (RunStopDeviceDiskTempVersionMode), 3001 (RunWithReturnCreateDisk), 3002 (RunWithReturnCreateDiskStatus), 3003 (RunWithReturnMapDisk), 3004 (RunWithReturnRebalanceDevices), 3005 (RunWithReturnCreateMaintenanceVersion), 3006 (RunWithReturnImportDisk), 4001 (RunByteArrayInputImportDevices), 4002 (RunByteArrayInputImportOemLicenses), 5001 (RunByteArrayOutputArchiveAuditTrail), 5002 (RunByteArrayOutputExportOemLicenses), 6001 (SetAuthGroup), 6002 (SetCollection), 6003 (SetDevice), 6004 (SetDisk), 6005 (SetDiskLocator), 6006 (SetFarm), 6007 (SetFarmView), 6008 (SetServer), 6009 (SetServerBiosBootstrap), 6010 (SetServerBootstrap), 6011 (SetServerStore), 6012 (SetSite), 6013 (SetSiteView), 6014 (SetStore), 6015 (SetUserGroup), 6016 SetVirtualHostingPool, 6017 SetUpdateTask, 6018 SetDiskUpdateDevice, 7001 (SetListDeviceBootstraps), 7002 (SetListDeviceBootstrapsDelete), 7003 (SetListDeviceBootstrapsAdd), 7004 (SetListDeviceCustomProperty), 7005 (SetListDeviceCustomPropertyDelete), 7006 (SetListDeviceCustomPropertyAdd), 7007 (SetListDeviceDiskPrinters), 7008 (SetListDeviceDiskPrintersDelete), 7009 (SetListDeviceDiskPrintersAdd), 7010 (SetListDevicePersonality), 7011 (SetListDevicePersonalityDelete), 7012 (SetListDevicePersonalityAdd), 7013 (SetListDiskLocatorCustomProperty), 7014 (SetListDiskLocatorCustomPropertyDelete), 7015 (SetListDiskLocatorCustomPropertyAdd), 7016 (SetListServerCustomProperty), 7017 (SetListServerCustomPropertyDelete), 7018 (SetListServerCustomPropertyAdd), 7019 (SetListUserGroupCustomProperty), 7020 (SetListUserGroupCustomPropertyDelete), and 7021 (SetListUserGroupCustomPropertyAdd)

uint Attachments: An or'ed value that indicates if there are any details for this action. A value of 15 indicates that there are Children, Sibling, Parameters and Properties for the action. Values are: 0 (None), 1 (Children), 2 (Sibling), 4 (Parameters), and 8 (Properties) Default=0

Guid Guid or AuditActionId: GUID of the action.

string Domain: Domain of the user that performed the action. Max Length=255

Guid ObjectId: GUID of the object of the action. Default=00000000-0000-0000-0000-000000000000

string ObjectName: Name of the object of the action. Default="" Max Length=1000

Guid ParentId: GUID of the parent action (one that triggered this action) if one exists. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

string Path: Path of the object of the action. An example is Site\Collection for a Device. Default="" Max Length=101

Guid RootId: GUID of the root action (one that triggered this group of actions) if one exists. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

Guid SiteId: GUID of the Site for the object of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

Guid SubId: GUID of the Collection or Store of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

DateTime Time: Date/Time the action occurred down to the millisecond. Has the date and time including milliseconds. Default=Empty

uint Type: Type of object that action was performed on. Values are: 0 (Many), 1 (AuthGroup), 2 (Collection), 3 (Device), 4 (Disk), 5 (DiskLocator), 6 (Farm), 7 (FarmView), 8 (Server), 9 (Site), 10 (SiteView), 11 (Store), 12 (System), and 13 (UserGroup)

string UserName: User that performed the action. Max Length=255

PvsAuthGroup

Read/Write Fields

string Name or AuthGroupName: Name of the Active Directory or Windows Group. Max Length=450

string Description: User description. Default="" Max Length=250

Read-Only Fields

Guid Guid or AuthGroupId: Read-only GUID that uniquely identifies this AuthGroup.

uint Role: Role of the AuthGroup for a Collection. role can only be used with CollectionId or CollectionName. 300 is Collection Administrator, and 400 is Collection Operator. Default=999

PvsAuthGroupUsage

Read-Only Fields

Guid Guid or Id: GUID of the item. The item can be a Farm, Site or Collection. It will be 00000000-0000-0000-0000-000000000000 for Farm.

string Name: Name of the item. The item can be a Farm, Site or Collection.

uint Role: Role of the AuthGroup for the item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 400 is Collection Operator. Default=999

PvsCeipData

Read/Write Fields

uint Enabled: 1 if CEIP is enabled, otherwise 0. Min=0, Max=1

uint InProgress: 1 if an upload is currently in progress, otherwise 0. Default=0

DateTime NextUpload: Date and time next CEIP upload is due if enabled is 1. Default=Empty

uint OneTimeUpload: 1 to perform a one time upload. Default=0

Guid ServerId: ID of server that is currently uploading, null if InProgress is 0. Default=00000000-0000-0000-0000-000000000000

Read-Only Field

Guid Uuid: CEIP UUID.

PvsCisData

Read/Write Fields

string Password: Password of the user required to obtain the token. This is required only by Set and Add

string Path: Path where the last problem report bundle was saved Default="" Max Length=255

string UserName: Username used to obtain the token Default="" Max Length=255

Read-Only Fields

Guid Guid or CisDataId: CIS UUID

string UploadToken: Token for uploading bundles to CIS Default="" Max Length=10

PvsCollection

Read/Write Fields

uint AutoAddNumberLength: The maximum length of the Device Number for Auto Add. This length plus the AutoAddPrefix length plus the AutoAddSuffix length must be less than 16. Required that ((lenautoAddPrefix+lenautoAddSuffix)+AutoAddNumberLength)<=15. Min=3, Max=9, Default=4

string AutoAddPrefix: The string put before the Device Number for Auto Add. Default="" ASCII computer name characters no end digit Max Length=12

string AutoAddSuffix: The string put after the Device Number for Auto Add. Default="" ASCII computer name characters no begin digit Max Length=12

bool AutoAddZeroFill: True when zeros be placed before the Device Number up to the AutoAddNumberLength for Auto Add, false otherwise. Default=true

string Name or CollectionName: Name of the Collection. It is unique within the Site. Max Length=50

string Description: User description. Default="" Max Length=250

bool Enabled: True when Devices in the Collection can be booted, false otherwise. Default=true

uint LastAutoAddDeviceNumber: The Device Number of the last Auto Added Device. Default=0

Guid TemplateDeviceId: GUID of a Device in the Collection whose settings are used for initial values of new Devices. Not used with templateDeviceName. Default=00000000-0000-0000-0000-000000000000

string TemplateDeviceName: Name of a Device in the Collection whose settings are used for initial values of new Devices. Not used with TemplateDeviceId. Default=""

Read-Only Fields

uint ActiveDeviceCount: Read-only count of active Devices in this Collection. Default=0

Guid Guid or CollectionId: Read-only GUID that uniquely identifies this Collection.

uint DeviceCount: Read-only count of Devices in this Collection. Default=0

uint DeviceWithPVDCount: Read-only count of Devices with Personal vDisk in this Collection. Default=0

uint MakActivateNeededCount: Read-only count of active Devices that need MAK activation in this Collection. Default=0

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 400 is Collection Operator. Default=999

Guid SiteId: GUID of the Site that this Collection is a member of. It is not used with SiteName.

string SiteName: Name of the Site that this Collection is a member of. It is not used with SiteId.

PvsConnection

Read/Write Fields

string Domain: Domain name to use for Authentication. If it has a value, it will be *****. Default=""

string Password: Password to use for Authentication. If it has a value, it will be *****. Default=""

string Persist: True when the connection settings should be, for Set, or have been, for Get, saved to the registry.

string Port: The Port to use to connect. Default=54321

string Name or Server: Name or IP of the Server to connect to. Default=localhost

string User: User name to use for Authentication. If it has a value, it will be *****. Default=""

Read-Only Field

string Connected: True when the Citrix.PVS.SnapIn is currently connected to the SoapServer with the settings in this PvsConnection.

PvsConnection can be created or modified using methods below:

New-Object Citrix.PVS.SnapIn.PvsConnection: Creates default Server=localhost, Port=54321, and no authentication.

New-Object Citrix.PVS.SnapIn.PvsConnection(Citrix.PVS.SnapIn copyFrom): Creates with settings of the copyFrom Citrix.PVS.SnapIn.

SetServerToLocalHostDefaultSettings: Server=localhost, Port=54321, and no authentication.

Copy(Citrix.PVS.SnapIn copyFrom): Modifies the settings to match the copyFrom Citrix.PVS.SnapIn.

Equals(Citrix.PVS.SnapIn compareTo): Returns true when the settings match what is in the compareTo.

PvsDevice

Read/Write Fields

string AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" ASCII Max Length=256

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

uint Authentication: Device log in authentication. Choices are 0 for none, 1 for User Name/Password, and 2 for Extern. This cannot be Set for a Device with Personal vDisk. Min=0, Max=2, Default=0

bool BdmBoot: Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false

DateTime BdmCreated: Timstamp when BDM device was created Default=Empty

uint BdmFormat: 1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0

uint BdmType: Use PXE boot when set to 0, BDM (Bios) boot when set to 1 and BDM (Uefi) boot when set to 2. Default=0

DateTime BdmUpdated: Timestamp of the last BDM boot disk update. Default=Empty

uint BootFrom: Device to boot from. Choices are 1 for vDisk, 2 for Hard Disk, and 3 for Floppy. This cannot be Set for a Device with Personal vDisk. Min=1, Max=3, Default=1

string ClassName: Used by Automatic Update feature to match new versions of Disks to a Device. This cannot be Set for a Device with Personal vDisk. Default="" Max Length=41

string Description: User description. Default="" Max Length=250

PvsPhysicalAddress DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device.

string Name or DeviceName: Computer name with no spaces. ASCII computer name characters Max Length=15

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

string DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

bool Enabled: True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true

bool LocalDiskEnabled: If there is a local disk menu choice for the Device, this is true. This cannot be Set for a Device with Personal vDisk. Default=false

uint LocalWriteCacheDiskSize: The size in GB to format the Device cache file disk. If the value is 0, then the disk is not formatted. Min=0, Max=2048, Default=0

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

string Password: Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=100

uint Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

uint Type: 1 when it performs test of Disks, 2 when it performs maintenance on Disks, 3 when it has a Personal vDisk, 4 when it has a Personal vDisk and performs tests, 0 otherwise. A Device with type 0 - 3 can only be Set to 0 - 3, and a Device with type 3 - 4 can only be Set to 3 - 4. Min=0, Max=4, Default=0

string User: Name of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=20

Guid XsPvsProxyUuid: UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000

Read-Only Fields

bool Active: True if the Device is currently active, false otherwise. Default=false

Guid CollectionId: GUID of the Collection this Device is to be a member of. It is not used with CollectionName.

string CollectionName: Name of the Collection this Device is to be a member of. SiteName or SiteId must also be used.

Guid Guid or DeviceId: Read-only GUID that uniquely identifies this Device.

string HypVmId: Hypervisor VM ID for HCL Default="" Max Length=250

string PvdDriveLetter: Read-only Personal vDisk Drive letter. Range is E to U and W to Z. Default="" Max Length=1

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 400 is Collection Operator. Default=999

Guid SiteId: GUID of the Site the CollectionName is to be a member of. This or SiteName is used with CollectionName.

string SiteName: Name of the Site the CollectionName is to be a member of. This or SiteId is used with CollectionName.

bool Template: True if the Device is the template in its Collection, false otherwise. Default=false

bool TemporaryVersionSet: Read-only true when temporary version is set. Default=false

Guid VirtualHostingPoolId: GUID that uniquely identifies the Virtual Hosting Pool for a VM. This is needed when Adding a VM device. Default=00000000-0000-0000-0000-000000000000

PvsDeviceBootstrap

Read-Only Fields

Guid Guid or DeviceId: GUID of the Device.

PvsDeviceBootstrapList: Each one of these has these 2 fields:

string Name or Bootstrap: Name of the bootstrap file. Max Length=259

string MenuText: Text that is displayed in the Boot Menu. If this field has no value, the bootstrap value is used. Default="" ASCII Max Length=64

PvsDeviceBootstrapList[] DeviceBootstrap: List of objects that can be changed with the methods below.

Methods

Add(string bootstrap, string menuText): Used to add a PvsDeviceBootstrapList to the end of the array.

Insert(int position, string bootstrap, string menuText): Used to insert a PvsDeviceBootstrapList array item at position. The position is 0 based.

Remove(int position): Used to remove a PvsDeviceBootstrapList array item at position. The position is 0 based.

Set(int position, string menuText): Used to set a PvsDeviceBootstrapList array item MenuText at position. The position is 0 based.

Reorder(int oldPosition, int newPosition): Used to move a PvsDeviceBootstrapList array item from the oldPosition to the newPosition. The oldPosition and newPosition are 0 based.

PvsDeviceBootstrapList

Read/Write Fields

string Name or Bootstrap: Name of the bootstrap file. Max Length=259

string MenuText: Text that is displayed in the Boot Menu. If this field has no value, the bootstrap value is used. Default="" ASCII Max Length=64

PvsDeviceDiskTempVersion

Read-Only Fields

Guid Guid or DeviceId: Read-only GUID that uniquely identifies the Device with temporary version.

string Name or DeviceName: Read-only Computer name that uniquely identifies the Device with temporary version. ASCII computer name characters

Guid DiskLocatorId: Read-only GUID that uniquely identifies then Disk Locator with temporary version.

string DiskLocatorName: Read-only Name of the Disk Locator File with temporary version. It is unique within the Store. ASCII

Guid SiteId: Read-only GUID of the Site the Device and DiskLocator are a member of.

string SiteName: Read-only Name of the Site the Device and DiskLocator are a member of.

Guid StoreId: Read-only GUID of the Store that the Disk Locator is a member of.

string StoreName: Read-only Name of the Store that the Disk Locator is a member of.

uint Version: Read-only Disk version the temporary is for.

PvsDeviceInfo

Read-Only Fields

bool Active: True if the Device is currently active, false otherwise. Default=false

string AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" ASCII Max Length=256

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

uint Authentication: Device log in authentication. Choices are 0 for none, 1 for User Name/Password, and 2 for Extern. This cannot be Set for a Device with Personal vDisk. Min=0, Max=2, Default=0

bool BdmBoot: Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false

DateTime BdmCreated: Timstamp when BDM device was created Default=Empty

uint BdmFormat: 1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0

uint BdmType: Use PXE boot when set to 0, BDM (Bios) boot when set to 1 and BDM (Uefi) boot when set to 2. Default=0

DateTime BdmUpdated: Timestamp of the last BDM boot disk update. Default=Empty

uint BootFrom: Device to boot from. Choices are 1 for vDisk, 2 for Hard Disk, and 3 for Floppy. This cannot be Set for a Device with Personal vDisk. Min=1, Max=3, Default=1

string ClassName: Used by Automatic Update feature to match new versions of Disks to a Device. This cannot be Set for a Device with Personal vDisk. Default="" Max Length=41

Guid CollectionId: GUID of the Collection this Device is to be a member of. It is not used with CollectionName.

string CollectionName: Name of the Collection this Device is to be a member of. SiteName or SiteId must also be used.

string Description: User description. Default="" Max Length=250

Guid Guid or DeviceId: Read-only GUID that uniquely identifies this Device.

PvsPhysicalAddress DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device.

string Name or DeviceName: Computer name with no spaces. ASCII computer name characters Max Length=15

string DiskFileName: Name of the Disk File including the extension. It is equal to "" if the Device is not active.

Guid DiskLocatorId: Read-only GUID of the Disk Locator that the Device is using. It is equal to 00000000-0000-0000-0000-000000000000 if the Device is not active.

string DiskLocatorName: Read-only name of the Disk Locator File that the Device is using. It is equal to the list of Disk Locator names for the Device if the Device is not active.

uint DiskVersion: Read-only version of the Disk Locator File that the Device is using. It is equal to 0 if the Device is not active. Default=0

uint DiskVersionAccess: State of the Disk Version. Values are: 0 (Production), 1 (Maintenance), 2 (MaintenanceHighestVersion), 3 (Override), 4 (Merge), 5 (MergeMaintenance), 6 (MergeTest), and 7 (Test). It is equal to 0 if the Device is not active. Default=0

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

string DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

bool Enabled: True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true

string HypVmId: Hypervisor VM ID for HCL Default="" Max Length=250

System.Net.IPAddress Ip: Read-only IP of the Device. It is equal to 0.0.0.0 if the Device is not active.

uint License: Oem Only: Read-only type of the license. Values are 0 when None, 1 or 2 when Desktop. It is equal to 0 if the Device is not active. Default=0

uint LicenseType: 0 when None, 1 for Desktop, 2 for Server, 5 for OEM SmartClient, 6 for XenApp, 7 for XenDesktop. It is equal to 0 if the Device is not active. Default=0

bool LocalDiskEnabled: If there is a local disk menu choice for the Device, this is true. This cannot be Set for a Device with Personal vDisk. Default=false

uint LocalWriteCacheDiskSize: The size in GB to format the Device cache file disk. If the value is 0, then the disk is not formatted. Min=0, Max=2048, Default=0

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

uint MakLicenseActivated: Read-only indicator if MAK licensing is being used and is activated. Values are: 0 (MAK not used), 1 (Not Activated), 2 (Activated). It is equal to 0 if the Device is not active. Default=0

string Model: Oem Only: Read-only model of the computer. Values are OptiPlex 745, 755, 320, 760, FX160, or Default. It is equal to "" if the Device is not active.

string Password: Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=100

uint Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

string PvdDriveLetter: Read-only Personal vDisk Drive letter. Range is E to U and W to Z. Default="" Max Length=1

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 400 is Collection Operator. Default=999

Guid ServerId: Read-only GUID of the Server that the Device is using. It is equal to 00000000-0000-0000-0000-000000000000 if the Device is not active.

System.Net.IPAddress ServerIpConnection: Read-only IP of the Server that the Device is using. It is equal to 0.0.0.0 if the Device is not active.

string ServerName: Read-only Name of the Server that the Device is using. It is equal to "" if the Device is not active.

uint ServerPortConnection: Read-only Port of the Server that the Device is using. It is equal to 0 if the Device is not active. Default=0

Guid SiteId: GUID of the Site the CollectionName is to be a member of. This or SiteName is used with CollectionName.

string SiteName: Name of the Site the CollectionName is to be a member of. This or SiteId is used with CollectionName.

string Status: 1 or 2 numbers in the format n,n. They are the number of retries and if ram cache is being used, ram cache percent used. It is equal to "" if the Device is not active.

bool Template: True if the Device is the template in its Collection, false otherwise. Default=false

bool TemporaryVersionSet: Read-only true when temporary version is set. Default=false

uint Type: 1 when it performs test of Disks, 2 when it performs maintenance on Disks, 3 when it has a Personal vDisk, 4 when it has a Personal vDisk and performs tests, 0 otherwise. Min=0, Max=4, Default=0

string User: Name of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=20

Guid VirtualHostingPoolId: GUID that uniquely identifies the Virtual Hosting Pool for a VM. This is needed when Adding a VM device. Default=00000000-0000-0000-0000-000000000000

Guid XsPvsProxyUuid: UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000

PvsDevicePersonality

Read-Only Fields

Guid Guid or DeviceId: GUID of the Device.

PvsDevicePersonalityList: Each one of these has these 2 fields:

string Name: Name of the Device personality item. Max Length=250

string Value: Value for the Device personality item. Max Length=1000

PvsDevicePersonalityList[] DevicePersonality: List of objects that can be changed with the methods below.

Methods

Add(string name, string value): Used to add a PvsDevicePersonalityList to the end of the array.

Insert(int position, string name, string value): Used to insert a PvsDevicePersonalityList array item at position. The position is 0 based.

Remove(int position): Used to remove a PvsDevicePersonalityList array item at position. The position is 0 based.

Set(int position, string value): Used to set a PvsDevicePersonalityList array item Value at position. The position is 0 based.

Reorder(int oldPosition, int newPosition): Used to move a PvsDevicePersonalityList array item from the oldPosition to the newPosition. The oldPosition and newPosition are 0 based.

PvsDevicePersonalityList

Read/Write Fields

string Name: Name of the Device personality item. Max Length=250

string Value: Value for the Device personality item. Max Length=1000

PvsDeviceStatus

Read-Only Fields

Guid Guid or DeviceId: Read-only GUID of the Device. Can be used with Get Device.

string Name or DeviceName: Read-only Name of the Device. Can be used with Get Device.

string DiskFileName: Name of the Disk File including the extension.

Guid DiskLocatorId: Read-only GUID of the Disk Locator that the Device is using.

string DiskLocatorName: Read-only name of the Disk Locator File that the Device is using.

uint DiskVersion: Read-only version of the Disk Locator File that the Device is using. Default=0

uint DiskVersionAccess: State of the Disk Version. Values are: 0 (Production), 1 (Maintenance), 2 (MaintenanceHighestVersion), 3 (Override), 4 (Merge), 5 (MergeMaintenance), 6 (MergeTest), and 7 (Test) Default=0

System.Net.IPAddress Ip: Read-only IP of the Device.

uint LicenseType: 0 when None, 1 for Desktop, 2 for Server, 5 for OEM SmartClient, 6 for XenApp, 7 for XenDesktop. Default=0

uint MakLicenseActivated: Read-only indicator if MAK licensing is being used and is activated. Values are: 0 (MAK not used), 1 (Not Activated), 2 (Activated). Default=0

Guid ServerId: Read-only GUID of the Server that the Device is using.

System.Net.IPAddress ServerIpConnection: Read-only IP of the Server that the Device is using.

string ServerName: Read-only Name of the Server that the Device is using.

uint ServerPortConnection: Read-only Port of the Server that the Device is using. Default=0

string Status: 1 or 2 numbers in the format n,n. They are the number of retries and if ram cache is being used, ram cache percent used.

PvsDisk

Read/Write Fields

bool ActivationDateEnabled: Use activation date to activate image when set to true. Default false

DateTime ActiveDate: Date to activate the disk if AutoUpdateEnabled and activationDateEnabled are true. Has the date. Empty when the AutoUpdateEnabled or activationDateEnabled are false.

bool AdPasswordEnabled: Enable AD password management when set to true.

string Author: User defined author. Max Length=40

bool AutoUpdateEnabled: Automatically update this image for matching Devices when set to true. Default false

UInt64 Build: User defined build number. Min=0, Max=4294967295, Default=0

string Class: Class of the Disk. Max Length=40

string ClearCacheDisabled: Clear cached secrets disabled.

string Company: User defined company. Max Length=40

string Date: User defined date. Max Length=40

bool VHDX: If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false

bool HaEnabled: Enable HA when set to true.

string HardwareTarget: User defined hardware target. Max Length=127

string ImageType: Type of this image (software type). Max Length=40

string InternalName: User defined name. Max Length=63

uint LicenseMode: 0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0

string LongDescription: Description of the Disk. Max Length=399

UInt64 MajorRelease: User defined major release number. Min=0, Max=4294967295, Default=0

UInt64 MinorRelease: User defined minor release number. Min=0, Max=4294967295, Default=0

string OperatingSystem: Operating System of Disk. Max Length=250

string OriginalFile: User defined original file. Max Length=127

string OsType: Operating System Type of Disk. Max Length=40

bool PrinterManagementEnabled: Invalid printers will be deleted from the Device when set to true.

string SerialNumber: User defined serial number. Max Length=36

string Title: User defined title. Max Length=40

UInt64 WriteCacheSize: RAM cache size (MB). Not 0 when used with Cache in Device RAM, and Cache in Device RAM with Overflow on Hard Disk. A value of 0 will disable the RAM use for Cache in Device RAM with Overflow on Hard Disk. Min=0, Max=131072, Default=0

uint WriteCacheType: 0 (Private), (other values are standard image) 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), or 9 (Cache in Device RAM with Overflow on Hard Disk). Min=0, Max=9, Default=0

Read-Only Fields

Guid Guid or DiskLocatorId: GUID of the DiskLocator used for Get and Set.

UInt64 DiskSize: Read-only size of the image. The value is 0 when it is not available. Default=0

string LogicalSectorSize: Logical Sector Size. Values are: 512, 4096, Default=512

uint VhdBlockSize: Block size in KB. For VHD it is only used with Dynamic type. Tested sizes for VHD are 512, 2048, and 16384. VHD Min=512, Max=16384, Default=2048. For VHDX it is used for all types. Tested size for VHDX is 32768. VHDX Min=1024, Max= 262144, Default=32768. Default=0

PvsDiskInfo

Read-Only Fields

bool ActivationDateEnabled: Use activation date to activate image when set to true. Default false

bool Active: True if the DiskLocator is currently active, false otherwise. Default=false

DateTime ActiveDate: Date to activate the disk if AutoUpdateEnabled and activationDateEnabled are true. Has the date. Empty when the AutoUpdateEnabled or activationDateEnabled are false.

bool AdPasswordEnabled: Enable AD password management when set to true.

string Author: User defined author. Max Length=40

bool AutoUpdateEnabled: Automatically update this image for matching Devices when set to true. Default false

UInt64 Build: User defined build number. Min=0, Max=4294967295, Default=0

string Class: Class of the Disk. Max Length=40

string ClearCacheDisabled: Clear cached secrets disabled.

string Company: User defined company. Max Length=40

string Date: User defined date. Max Length=40

string Description: User description. Default="" Max Length=250

uint DeviceCount: Read-only count of Devices. Default=0

Guid Guid or DiskLocatorId: Read-only GUID that uniquely identifies this Disk Locator.

string Name or DiskLocatorName: Name of the Disk Locator File. It is unique within the Store. ASCII Max Length=52

UInt64 DiskSize: Read-only size of the image. The value is 0 when it is not available. Default=0

Guid DiskUpdateDeviceId: GUID of the DiskUpdateDevice that is used when updates are performed. Default=00000000-0000-0000-0000-000000000000

string DiskUpdateDeviceName: Name of the DiskUpdateDevice that is used when updates are performed. Default=""

bool Enabled: True when this disk can be booted, false otherwise. Default=true

bool EnabledForDevice: True when this disk is enabled for the Device specified, false otherwise. This is only returned when a Device is specified. Default=true

bool VHDX: If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false

bool HaEnabled: Enable HA when set to true.

string HardwareTarget: User defined hardware target. Max Length=127

string ImageType: Type of this image (software type). Max Length=40

string InternalName: User defined name. Max Length=63

uint LicenseMode: 0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0

bool Locked: True if the Disk is currently locked, false otherwise. Default=false

string LogicalSectorSize: Logical Sector Size. Values are: 512, 4096, Default=512

string LongDescription: Description of the Disk. Max Length=399

UInt64 MajorRelease: User defined major release number. Min=0, Max=4294967295, Default=0

bool Mapped: True if the Disk is currently mapped, false otherwise. Default=false

string MenuText: Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default="" ASCII Max Length=64

UInt64 MinorRelease: User defined minor release number. Min=0, Max=4294967295, Default=0

string OperatingSystem: Operating System of Disk. Max Length=250

string OriginalFile: User defined original file. Max Length=127

string OsType: Operating System Type of Disk. Max Length=40

bool PrinterManagementEnabled: Invalid printers will be deleted from the Device when set to true.

bool RebalanceEnabled: True when this Server can automatically rebalance Devices, false otherwise. Default=false

uint RebalanceTriggerPercent: Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 999 is read-only. Default=999

string SerialNumber: User defined serial number. Max Length=36

Guid ServerId: GUID of the single Server that this Disk Locator is assigned to. It is not used with ServerName. Default=00000000-0000-0000-0000-000000000000

string ServerName: Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=""

Guid SiteId: GUID of the Site this DiskLocator is to be a member of. It is not used with SiteName.

string SiteName: Name of the Site this DiskLocator is to be a member of. It is not used with SiteId.

Guid StoreId: GUID of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreName.

string StoreName: Name of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreId.

uint SubnetAffinity: Qualifier for subnet affinity when assigning a Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2, Default=0

bool TemporaryVersionSet: Read-only true when temporary version(s) are set. Default=false

string Title: User defined title. Max Length=40

uint VhdBlockSize: Block size in KB. For VHD it is only used with Dynamic type. Tested sizes for VHD are 512, 2048, and 16384. VHD Min=512, Max=16384, Default=2048. For VHDX it is used for all types. Tested size for VHDX is 32768. VHDX Min=1024, Max= 262144, Default=32768. Default=0

UInt64 WriteCacheSize: RAM cache size (MB). Not 0 when used with Cache in Device RAM, and Cache in Device RAM with Overflow on Hard Disk. A value of 0 will disable the RAM use for Cache in Device RAM with Overflow on Hard Disk. Min=0, Max=131072, Default=0

uint WriteCacheType: 0 (Private), (other values are standard image) 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), or 9 (Cache in Device RAM with Overflow on Hard Disk). Min=0, Max=9, Default=0

PvsDiskInventory

Read-Only Fields

string Active: 1 if the Server is currently active, 2 if unknown, and 0 otherwise.

Guid Guid or DiskLocatorId: GUID of the DiskLocator used for Get and Set.

string FilePath: Path used to access the disk version from the Server. Empty if the information is not available.

DateTime FileTime: Date/Time of the date version file. Has the date and time without milliseconds. Empty if the information is not available.

DateTime PropertiesTime: Date/Time of the disk properties. Has the date and time without milliseconds. Empty if the information is not available.

Guid ServerId: GUID of the Server that the Disk Version Inventory is being reported about.

string ServerName: Name of the Server that the Disk Version Inventory is being reported about.

string State: The number code of the inventory state. Values are: 0 (Up to date), 1 (version file is missing), 2 (version file is out of date), 3 (properties are missing), 4 (properties are out of date), 5 (server is not reachable).

string Version: Version number. The base disk is version 0, the other version numbers are in part of the file name.

PvsDiskLocator

Read/Write Fields

string Description: User description. Default="" Max Length=250

bool Enabled: True when this disk can be booted, false otherwise. Default=true

string MenuText: Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default="" ASCII Max Length=64

bool RebalanceEnabled: True when this Server can automatically rebalance Devices, false otherwise. Default=false

uint RebalanceTriggerPercent: Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25

Guid ServerId: GUID of the single Server that this Disk Locator is assigned to. It is not used with ServerName. Default=00000000-0000-0000-0000-000000000000

string ServerName: Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=""

uint SubnetAffinity: Qualifier for subnet affinity when assigning a Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2, Default=0

Read-Only Fields

bool Active: True if the DiskLocator is currently active, false otherwise. Default=false

Guid Guid or DiskLocatorId: Read-only GUID that uniquely identifies this Disk Locator.

string Name or DiskLocatorName: Name of the Disk Locator File. It is unique within the Store. ASCII Max Length=52

Guid DiskUpdateDeviceId: GUID of the DiskUpdateDevice that is used when updates are performed. Default=00000000-0000-0000-0000-000000000000

string DiskUpdateDeviceName: Name of the DiskUpdateDevice that is used when updates are performed. Default=""

bool EnabledForDevice: True when this disk is enabled for the Device specified, false otherwise. This is only returned when a Device is specified. Default=true

bool Mapped: True if the Disk is currently mapped, false otherwise. Default=false

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 999 is read-only. Default=999

Guid SiteId: GUID of the Site this DiskLocator is to be a member of. It is not used with SiteName.

string SiteName: Name of the Site this DiskLocator is to be a member of. It is not used with SiteId.

Guid StoreId: GUID of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreName.

string StoreName: Name of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreId.

bool TemporaryVersionSet: Read-only true when temporary version(s) are set. Default=false

PvsDiskLocatorLock

Read-Only Fields

Guid DeviceId: GUID of the Device that has the lock, will be 00000000-0000-0000-0000-000000000000 if a Server has the lock.

string DeviceName: Name of the Device that has the lock, will not be included if a Server has the lock.

bool Exclusive: True when the lock is exclusive, false when it is shared. Default=false

bool ReadOnly: True when lock is because file system is read only, false when file system is read write Default=false

Guid ServerId: GUID of the Server that has the lock, will be 00000000-0000-0000-0000-000000000000 if a Device has the lock.

string ServerName: Name of the Server that has the lock, will not be included if a Device has the lock.

PvsDiskUpdateDevice

Read/Write Fields

string AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" Max Length=256

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

string Description: User description. Default="" Max Length=250

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

string DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

uint Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

Read-Only Fields

bool Active: True if the Device is currently active, false otherwise. Default=false

Guid Guid or DeviceId: Read-only GUID that uniquely identifies this Device.

PvsPhysicalAddress DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device.

string Name or DeviceName: Computer name with no spaces. ASCII computer name characters Max Length=15

Guid DiskLocatorId: GUID of the Disk Locator to update with this Device.

string DiskLocatorName: Name of the Disk Locator File to update with this Device.

uint DiskVersion: Read-only version of the Disk Locator File that the Device is using. It is equal to 0 if the Device is not active. Default=0

System.Net.IPAddress Ip: Read-only IP of the Device. It is equal to 0.0.0.0 if the Device is not active.

uint License: Oem Only: Read-only type of the license. Values are 0 when None, 1 or 2 when Desktop. It is equal to 0 if the Device is not active. Default=0

uint LicenseType: 0 when None, 1 for Desktop, 2 for Server, 5 for OEM SmartClient, 6 for XenApp, 7 for XenDesktop. It is equal to 0 if the Device is not active. Default=0

uint MakLicenseActivated: Read-only indicator if MAK licensing is being used and is activated. Values are: 0 (MAK not used), 1 (Not Activated), 2 (Activated). It is equal to 0 if the Device is not active. Default=0

string Model: Oem Only: Read-only model of the computer. Values are OptiPlex 745, 755, 320, 760, FX160, or Default. It is equal to "" if the Device is not active.

Guid ServerId: Read-only GUID of the Server that the Device is using. It is equal to 00000000-0000-0000-0000-000000000000 if the Device is not active.

System.Net.IPAddress ServerIpConnection: Read-only IP of the Server that the Device is using. It is equal to 0.0.0.0 if the Device is not active.

string ServerName: Read-only Name of the Server that the Device is using. It is equal to "" if the Device is not active.

uint ServerPortConnection: Read-only Port of the Server that the Device is using. It is equal to 0 if the Device is not active. Default=0

Guid SiteId: GUID of the Site this Disk Update Device is to be a member of.

string SiteName: Name of the Site this Disk Update Device is to be a member of.

string Status: 1 or 2 numbers in the format n,n. They are the number of retries and if ram cache is being used, ram cache percent used. It is equal to "" if the Device is not active.

Guid StoreId: GUID of the Store that the Disk Locator is a member of.

string StoreName: Name of the Store that the Disk Locator is a member of.

Guid VirtualHostingPoolId: GUID of the Virtual Hosting Pool. It is not used with VirtualHostingPoolName. Default=00000000-0000-0000-0000-000000000000

string VirtualHostingPoolName: Name of the Virtual Hosting Pool.

PvsDiskUpdateStatus

Read-Only Fields

uint CurrentStatus: Current status of the update. Values are: 0 (Ready), 1 (Update Pending), 2 (Preparing Image), 3 (Starting VM), 4 (Update In Progress), 5 (Stopping VM), 6 (Submitting Image), 7 (Reverting Image), 8 (Invalid), 9 (Aborted), 10 (Completed Successfully), 11 (No Updates) Min=0, Max=11, Default=0

string CurrentStatusMessage: Message string that includes the results of the run. Default="" Max Length=255

string Description: User description of the Update Task.

Guid DeviceId: GUID that Device being used to do the update.

string DeviceName: Name of the Device being used to do the update.

Guid DiskLocatorId: GUID of the Disk Locator to update.

string Name or DiskLocatorName: Name of the Disk Locator File to update.

Guid Guid or DiskUpdateTaskId: GUID that uniquely identifies this Update Task and Device relationship.

uint PreviousResult: Status of the last run. Values are: 0 (Ready), 1 (Update Pending), 2 (Preparing Image), 3 (Starting VM), 4 (Update In Progress), 5 (Stopping VM), 6 (Submitting Image), 7 (Reverting Image), 8 (Invalid), 9 (Aborted), 10 (Completed Successfully), 11 (No Updates) Min=0, Max=11, Default=0

string PreviousResultMessage: Message string that includes the results of the last run. Default="" Max Length=255

Guid SiteId: GUID of the Site that this Update Task Name is a member of.

string SiteName: Name of the Site that this Update Task Name is a member of.

Guid StoreId: GUID of the Store that the Disk Locator is a member of.

string StoreName: Name of the Store that the Disk Locator is a member of.

Guid UpdateTaskId: GUID that uniquely identifies the Update Task.

string UpdateTaskName: Name of the Update Task.

Guid VirtualHostingPoolId: GUID of the Virtual Hosting Pool being used for the update.

string VirtualHostingPoolName: Name of the Virtual Hosting Pool being used for the update.

PvsDiskVersion

Read/Write Fields

string Description: User description. Default="" Max Length=250

DateTime ScheduledDate: Date/Time that the Disk Version is scheduled to become available. Has the date, hour and minute. Empty when the disk version is made available immediately. Default=Empty

Read-Only Fields

uint Access: Read-only access of the Disk Version. Values are: 0 (Production), 1 (Maintenance), 2 (MaintenanceHighestVersion), 3 (Override), 4 (Merge), 5 (MergeMaintenance), 6 (MergeTest), and 7 (Test) Min=0, Max=7, Default=0

bool CanDelete: Read-only true when the version can be deleted. Default=false

bool CanMerge: Read-only true when the version can be update merged. Will be set for the highest version number. Default=false

bool CanMergeBase: Read-only true when the version can be base merged. Will be set for the highest version number. Default=false

bool CanOverride: Read-only true when the version can be set as the Override. Default=false

bool CanPromote: Read-only true when the version can be promoted. Default=false

bool CanRevertMaintenance: Read-only true when the version can be reverted to Maintenance Access. Default=false

bool CanRevertTest: Read-only true when the version can be reverted to Test Access. Default=false

bool CanSetScheduledDate: Read-only true when the version can have the scheduled date modified. Default=false

string CreateDate: Read-only Date/Time that the Disk Version was created. Default=getdate

bool DeleteWhenFree: Read-only true if the Disk Version is no longer needed because of a merge. If not current booted by a Device, it can be deleted. Default=false

uint DeviceCount: Read-only count of Devices. Default=0

string Name or DiskFileName: Name of the Disk File including the extension. Default=""

Guid Guid or DiskLocatorId: GUID of the DiskLocator used for Get and Set.

bool GoodInventoryStatus: True when the up to date file is accessible by all Servers, false otherwise. Default=false

bool IsPending: Read-only true when the version ScheduledDate has not occurred. Default=false

uint TaskId: When a Merge is occurring, this will be set with the task number of the process that is occurring. Default=""

bool TemporaryVersionSet: Read-only true when temporary version(s) are set. Some changes cannot be made to the version when this is set. Default=false

uint Type: Read-only type of the Disk Version. Values are: 0 (Base), 1 (Manual), 2 (Automatic), 3 (Merge), and 4 (MergeBase) Min=0, Max=4, Default=0

uint Version: Read-only version number. The base disk is version 0, the other version numbers are in part of the file name. Default=0

PvsFarm

Read/Write Fields

bool AuditingEnabled: True when Auditing is enabled, false otherwise. Default=false

bool AutoAddEnabled: True when Auto Add is enabled, false otherwise. Default=false

bool AutomaticMergeEnabled: True when Automatic Merge is enabled, false otherwise. If the number of versions becomes more than the MaxVersions value, a merge will occur at the end of PromoteDiskVersion. Default=true

Guid DefaultSiteId: GUID of the Site to place new Devices into automatically. Not used with defaultSiteName. Default=00000000-0000-0000-0000-000000000000

string DefaultSiteName: Name of the Site to place new Devices into automatically. Not used with DefaultSiteId. Default=""

string Description: User description. Default="" Max Length=250

string Name or FarmName: Name of the Farm. Default="" Max Length=50

DateTime LastAuditArchiveDate: Last date of Audit Trail data that was Archived. Has the date. Default=Empty

string LicenseServer: License server name. Default="" Max Length=255

uint LicenseServerPort: License server port. Min=1025, Max=65534, Default=27000

bool LicenseTradeUp: License server trade up, when set to true. Default=false

uint MaxVersions: Maximum number a versions of a Disk that can exist before a merge will automatically occur. Min=3, Max=50, Default=5

uint MergeMode: Mode to place the version in after a merge has occurred. Values are: 0 (Production), 1 (Test) and 2 (Maintenance). Min=0, Max=2, Default=2

bool OfflineDatabaseSupportEnabled: True when Offline Database Support is enabled, false otherwise. Default=false

Read-Only Fields

bool AdGroupsEnabled: Active Directory groups are used for authorization, when set to true. Windows groups are used when set to false. Default=false

string DatabaseInstanceName: Read-only name of the database instance.

string DatabaseName: Read-only name of the database.

string DatabaseServerName: Read-only name of the database server.

string FailoverPartnerInstanceName: Read-only name of the database server instance.

string FailoverPartnerServerName: Read-only name of the database server.

Guid Guid or FarmId: Read-only GUID that uniquely identifies this Farm.

string MultiSubnetFailover: Read-only Database MultiSubnetFailover value

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, and 999 is read-only. Default=999

PvsFarmView

Read/Write Fields

string Description: User description. Default="" Max Length=250

string Name or FarmViewName: name of the Farm View. Max Length=50

Read-Only Fields

uint ActiveDeviceCount: Read-only count of active Devices in this Farm View. Default=0

uint DeviceCount: Read-only count of Devices in this Farm View. Default=0

Guid Guid or FarmViewId: Read-only GUID that uniquely identifies this Farm View.

uint MakActivateNeededCount: Read-only count of active Devices that need MAK activation in this Farm View. Default=0

PvsGroup

Read-Only Fields

Guid Guid: GUID of the Active Directory group. 00000000-0000-0000-0000-000000000000 for Windows groups.

string Name: Name of the Group.

PvsLocalServer

Read-Only Field

string Name or LocalServer: NetBios name of local server.

PvsNewDiskVersion

Read-Only Fields

string Name: Name of the disk file without the extension.

uint Status: Status of the disk file. Values are: 0 (Valid), 1 (Missing Properties File), 2 (Access Denied), 3 (Access Denied and Missing Properties File), 4 (Invalid Disk File), 5 (Manifest Invalid)

PvsPhysicalAddress

Derived from System.Net.NetworkInformation.PhysicalAddress. GetString() returns a - delimited MAC address.

PvsServer

Read/Write Fields

uint AdMaxPasswordAge: Number of days before a password expires. Min=1, Max=30, Default=7

bool AdMaxPasswordAgeEnabled: Age the password, when set to true. Default=false

uint BootPauseSeconds: Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10

uint BuffersPerThread: Number of buffers per worker thread. Min=1, Max=128, Default=24

uint BusyDbConnectionRetryCount: Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2

uint BusyDbConnectionRetryInterval: Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25

string Description: User description. Default="" Max Length=250

bool EventLoggingEnabled: Enable event logging, when set to true. Default=false

uint FirstPort: Number of the first UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6910

uint InitialQueryConnectionPoolSize: Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50

uint InitialTransactionConnectionPoolSize: Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50

uint IoBurstSize: Number of bytes read/writes can send in a burst of packets. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=4096, Max=61440, Default=32768

System.Net.IPAddress[] Ip: One or more streaming ip addresses.

DateTime LastBugReportAttempt: Time that this server last attempted to upload or generate a bug report bundle. Default=Empty

string LastBugReportResult: Status of the last bug report on this server. Default="" Max Length=4000

string LastBugReportStatus: Status of the last bug report on this server. Default="" Max Length=250

string LastBugReportSummary: Summary of the last bug report on this server. Default="" Max Length=250

DateTime LastCeipUploadAttempt: Time that this server last attempted a CEIP upload. Default=Empty

uint LastPort: Number of the last UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6930

uint LicenseTimeout: Amount of seconds before a license times out. Min=15, Max=300, Default=30

uint LocalConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are local. A value of 0 disables the feature. Min=0, Max=128, Default=4

uint LogFileBackupCopiesMax: Maximum number of log file backups. Min=1, Max=50, Default=4

uint LogFileSizeMax: Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=4

uint MaxBootDevicesAllowed: Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500

uint MaxBootSeconds: Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60

uint MaxQueryConnectionPoolSize: Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000

uint MaxTransactionConnectionPoolSize: Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000

uint MaxTransmissionUnits: Ethernet maximum transmission unit size for the protocol for use for Server and Device. Required that

IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=502, Max=16426, Default=1506

bool NonBlockingIoEnabled: Use non-Blocking IO, when set to true. Default=true

float PowerRating: A strictly relative rating of this Server's capabilities when compared to other Servers in the Store(s) it belongs too; can be used to help tune load balancing. Min=0.1, Max=1000, Default=1

uint RefreshInterval: Interval, in number of seconds, the server should wait before refreshing settings. If set to 0, unused database connections are never released. Min=0, Max=32767, Default=300

uint RemoteConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4

uint ServerCacheTimeout: Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8

string Name or ServerName: Computer name with no spaces. ASCII computer name characters Max Length=21

uint ThreadsPerPort: Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8

uint UnusedDbConnectionTimeout: Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300

uint VDiskCreatePacing: VDisk create time pacing in miliseconds. Min=0, Max=5, Default=0

Read-Only Fields

uint Active: 1 if the Server is currently active, 2 if unknown, and 0 otherwise. Min=0, Max=2, Default=0

System.Net.IPAddress ManagementIp: IP address used for management communications between Servers. Default=0.0.0.0

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, and 200 is Site Administrator. Default=999

string ServerFqdn: Read-only fully qualified domain name. Default="" Max Length=1024

Guid Guid or ServerId: Read-only GUID that uniquely identifies this Server.

Guid SiteId: GUID of the Site this Server is to be a member of. It is not used with SiteName.

string SiteName: Name of the Site this Server is to be a member of. It is not used with SiteId.

PvsServerBiosBootstrap

Read/Write Fields

bool BootFromHdOnFail: For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false

System.Net.IPAddress Bootserver1_Ip: 1st boot server IP. Only used when Lookup is false.

uint Bootserver1_Port: 1st boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

System.Net.IPAddress Bootserver2_Ip: 2nd boot server IP. Only used when Lookup is false. Default=0.0.0.0

uint Bootserver2_Port: 2nd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

System.Net.IPAddress Bootserver3_Ip: 3rd boot server IP. Only used when Lookup is false. Default=0.0.0.0

uint Bootserver3_Port: 3rd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

System.Net.IPAddress Bootserver4_Ip: 4th boot server IP. Only used when Lookup is false. Default=0.0.0.0

uint Bootserver4_Port: 4th boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

bool DhcpEnabled: Use DHCP to retrieve target device IP when set to true, otherwise use the static domain, dnsIpAddresstrue and dnsIpAddress2 settings. Default=true

System.Net.IPAddress DnsIpAddress1: Primary DNS server IP. Only used when DhcpEnabled is false.

System.Net.IPAddress DnsIpAddress2: Secondary DNS server IP. Only used when DhcpEnabled is false.

string Domain: Domain of the primary and secondary DNS servers. Only used when DhcpEnabled is false.

bool Enabled: Automatically update the BIOS on the target device with these setting when set to true, otherwise do not use these settings. Default=false

uint GeneralTimeout: Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000

bool InterruptSafeMode: Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false

bool Lookup: Use DNS to find the Server when set to true with the ServerName host value, otherwise use the bootservertrue_Ip, bootservertrue_Port, bootserver2_Ip, bootserver2_Port, bootserver3_Ip, bootserver3_Port, bootserver4_Ip, and bootserver4_Port settings. Default=true

bool PaeMode: PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false

uint PollingTimeout: Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000

uint RecoveryTime: When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50

string Name or ServerName: Host to use for DNS lookup. Only used when Lookup is true. Default=IMAGESERVER1

bool VerboseMode: Display verbose diagnostic information when set to true. Default=false

Read-Only Field

Guid Guid or ServerId: GUID of the Server used for Get and Set.

PvsServerBootstrap

Read/Write Fields

bool BootFromHdOnFail: For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false

System.Net.IPAddress Bootserver1_Gateway: 1st boot server gateway. Default=0.0.0.0

System.Net.IPAddress Bootserver1_Ip: 1st boot server IP.

System.Net.IPAddress Bootserver1_Netmask: 1st boot server netmask. Default=0.0.0.0

uint Bootserver1_Port: 1st boot server port. Min=1025, Max=65536, Default=6910

System.Net.IPAddress Bootserver2_Gateway: 2nd boot server gateway. Default=0.0.0.0

System.Net.IPAddress Bootserver2_Ip: 2nd boot server IP. Default=0.0.0.0

System.Net.IPAddress Bootserver2_Netmask: 2nd boot server netmask. Default=0.0.0.0

uint Bootserver2_Port: 2nd boot server port. Min=1025, Max=65536, Default=6910

System.Net.IPAddress Bootserver3_Gateway: 3rd boot server gateway. Default=0.0.0.0

System.Net.IPAddress Bootserver3_Ip: 3rd boot server IP. Default=0.0.0.0

System.Net.IPAddress Bootserver3_Netmask: 3rd boot server netmask. Default=0.0.0.0

uint Bootserver3_Port: 3rd boot server port. Min=1025, Max=65536, Default=6910

System.Net.IPAddress Bootserver4_Gateway: 4th boot server gateway. Default=0.0.0.0

System.Net.IPAddress Bootserver4_Ip: 4th boot server IP. Default=0.0.0.0

System.Net.IPAddress Bootserver4_Netmask: 4th boot server netmask. Default=0.0.0.0

uint Bootserver4_Port: 4th boot server port. Min=1025, Max=65536, Default=6910

uint GeneralTimeout: Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000

bool InterruptSafeMode: Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false

bool PaeMode: PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false

uint PollingTimeout: Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000

uint RecoveryTime: When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50

bool VerboseMode: Display verbose diagnostic information when set to true. Default=false

Read-Only Fields

string Name: Name of the bootstrap file used to Get and Set.

Guid Guid or ServerId: GUID of the Server used for Get and Set.

PvsServerBootstrapName

Read-Only Field

string Name: Bootstrap file name.

PvsServerInfo

Read-Only Fields

uint Active: 1 if the Server is currently active, 2 if unknown, and 0 otherwise. Min=0, Max=2, Default=0

uint AdMaxPasswordAge: Number of days before a password expires. Min=1, Max=30, Default=7

bool AdMaxPasswordAgeEnabled: Age the password, when set to true. Default=false

uint BootPauseSeconds: Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10

uint BuffersPerThread: Number of buffers per worker thread. Min=1, Max=128, Default=24

uint BusyDbConnectionRetryCount: Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2

uint BusyDbConnectionRetryInterval: Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25

System.Net.IPAddress ContactIp: Read-only contact IP for the Server.

string ContactPort: Read-only contact port for the Server.

string Description: User description. Default="" Max Length=250

uint DeviceCount: Read-only count of Devices. Default=0

bool EventLoggingEnabled: Enable event logging, when set to true. Default=false

uint FirstPort: Number of the first UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6910

uint InitialQueryConnectionPoolSize: Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50

uint InitialTransactionConnectionPoolSize: Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50

uint IoBurstSize: Number of bytes read/writes can send in a burst of packets. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=4096, Max=61440, Default=32768

System.Net.IPAddress[] Ip: One or more streaming ip addresses.

DateTime LastBugReportAttempt: Time that this server last attempted to upload or generate a bug report bundle. Default=Empty

string LastBugReportResult: Status of the last bug report on this server. Default="" Max Length=4000

string LastBugReportStatus: Status of the last bug report on this server. Default="" Max Length=250

string LastBugReportSummary: Summary of the last bug report on this server. Default="" Max Length=250

DateTime LastCeipUploadAttempt: Time that this server last attempted a CEIP upload. Default=Empty

uint LastPort: Number of the last UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6930

uint LicenseTimeout: Amount of seconds before a license times out. Min=15, Max=300, Default=30

uint LocalConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are local. A value of 0 disables the feature. Min=0, Max=128, Default=4

uint LogFileBackupCopiesMax: Maximum number of log file backups. Min=1, Max=50, Default=4

uint LogFileSizeMax: Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=4

System.Net.IPAddress ManagementIp: IP address used for management communications between Servers. Default=0.0.0.0

uint MaxBootDevicesAllowed: Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500

uint MaxBootSeconds: Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60

uint MaxQueryConnectionPoolSize: Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000

uint MaxTransactionConnectionPoolSize: Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000

uint MaxTransmissionUnits: Ethernet maximum transmission unit size for the protocol for use for Server and Device. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=502, Max=16426, Default=1506

bool NonBlockingIoEnabled: Use non-Blocking IO, when set to true. Default=true

float PowerRating: A strictly relative rating of this Server's capabilities when compared to other Servers in the Store(s) it belongs too; can be used to help tune load balancing. Min=0.1, Max=1000, Default=1

uint RefreshInterval: Interval, in number of seconds, the server should wait before refreshing settings. If set to 0, unused database connections are never released. Min=0, Max=32767, Default=300

uint RemoteConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, and 200 is Site Administrator. Default=999

uint ServerCacheTimeout: Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8

string ServerFqdn: Read-only fully qualified domain name. Default="" Max Length=1024

Guid Guid or ServerId: Read-only GUID that uniquely identifies this Server.

string Name or ServerName: Computer name with no spaces. ASCII computer name characters Max Length=21

Guid SiteId: GUID of the Site this Server is to be a member of. It is not used with SiteName.

string SiteName: Name of the Site this Server is to be a member of. It is not used with SiteId.

uint ThreadsPerPort: Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8

uint UnusedDbConnectionTimeout: Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300

uint VDiskCreatePacing: VDisk create time pacing in miliseconds. Min=0, Max=5, Default=0

PvsServerStatus

Read-Only Fields

uint DeviceCount: Read-only count of Devices. Default=0

System.Net.IPAddress Ip: Read-only contact IP for the Server.

uint Port: Read-only contact port for the Server.

Guid Guid or ServerId: Read-only GUID of the Server. Can be used with Get Server.

string Name or ServerName: Read-only Name of the Server. Can be used with Get Server.

uint Status: Status of the server, 0 if down, 1 if up and 2 if unknown. Default=0

PvsServerStore

Read/Write Fields

string[] CachePath: Cache path(s) that the Server uses with the Store. If none are specified the caches will be placed in the Store cachePath. Default=None

string Path: Directory path that the Server uses to access the Store. Default="" Max Length=255

Read-Only Fields

Guid ServerId: GUID of the server that uses the Store. ServerName can be used instead.

string ServerName: Name of the server that uses the Store. ServerId can be used instead.

Guid StoreId: GUID of the Store. StoreName can be used instead.

string StoreName: Name of the Store. StoreId can be used instead.

PvsSite

Read/Write Fields

Guid DefaultCollectionId: GUID of the Collection to place new Devices into automatically. Not used with defaultCollectionName. Default=00000000-0000-0000-0000-000000000000

string DefaultCollectionName: Name of the Collection to place new Devices into automatically. Not used with DefaultCollectionId. Default=""

string Description: User description. Default="" Max Length=250

Guid DiskUpdateServerId: GUID of the Disk Update Server for the Site. Not used with DiskUpdateServerName. Default=00000000-0000-0000-0000-000000000000

string DiskUpdateServerName: Name of the Disk Update Server for the Site. Not used with DiskUpdateServerId. Default=""

bool EnableDiskUpdate: True when Disk Updated is enabled for the Site, false otherwise. Default=false

uint InventoryFilePollingInterval: The number of seconds between polls for Disk changes in the Stores. Min=1, Max=600, Default=60

string MakPassword: User password used for MAK activation. Default="" Max Length=64

string MakUser: User name used for MAK activation. Default="" Max Length=64

string Name or SiteName: Name of the Site. Max Length=50

Read-Only Fields

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, and 999 is read-only. Default=999

Guid Guid or SiteId: Read-only GUID that uniquely identifies this Site.

PvsSiteView

Read/Write Fields

string Description: User description. Default="" Max Length=250

string Name or SiteViewName: Name of the Site View. Max Length=50

Read-Only Fields

uint ActiveDeviceCount: Read-only count of active Devices in this Site View. Default=0

uint DeviceCount: Read-only count of Devices in this Site View. Default=0

uint DeviceWithPVDCount: Read-only count of Devices with Personal vDisk in this Site View. Default=0

uint MakActivateNeededCount: Read-only count of active Devices that need MAK activation in this Site View. Default=0

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, and 200 is Site Administrator. Default=999

Guid SiteId: GUID of the Site this View is to be a member of. It is not used with SiteName.

string SiteName: Name of the Site this View is to be a member of. It is not used with SiteId.

Guid Guid or SiteViewId: Read-only GUID that uniquely identifies this Site View.

PvsStore

Read/Write Fields

string[] CachePath: Default Cache path(s) that the Servers use with this Store. If none are specified the caches will be placed in the WriteCache subdirectory of the Store path. Default=None

string Description: User description. Default="" Max Length=250

string Path: Default directory path that the Servers use to access this Store. Max Length=255

Guid SiteId: GUID of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteName can be used instead. Default=00000000-0000-0000-0000-000000000000

string SiteName: Name of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteId can be used instead. Default=""

string Name or StoreName: Name of the Store. Max Length=50

Read-Only Fields

string PathType: Read-only field indicating if the vdisks are on a server's local hard disk or on a remote share.

uint Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, and 999 is read-only. Default=999

Guid Guid or StoreId: Read-only GUID that uniquely identifies this Store.

PvsStoreSharedOrServerPath

Read-Only Fields

string Path: Directory path that the Servers use to access this Store.

Guid StoreId: GUID of the Store.

string StoreName: Name of the Store.

PvsTask

Read-Only Fields

string Command: Command being processed. Default="" Max Length=50

string CommandType: Type of the command. Values are: Add, Delete, Get, Info, Run, RunWithReturn, Set and SetList. Default="" Max Length=13

DateTime ExpirationTime: Time the task record may be removed from the database if the task does not complete. Has the date and time without milliseconds.

uint Handle: Handle to a running function.

System.Net.IPAddress Ip: IP Address of the remote host.

string MapiException: Exception result in XML format. Default=""

uint Port: Port number of the remote service.

string Results: Result in XML format. Default=""

string ServerFqdn: Qualified name of the server. Default="" Max Length=1024

Guid SiteId: GUID of the Site that this Task is being processed in. Default=00000000-0000-0000-0000-000000000000

string SiteName: Name of the Site that that this Task is being processed in.

DateTime StartTime: Time the task was started. Has the date and time without milliseconds.

uint State: State of the Task. Values are: 0 (Processing), 1 (Cancelled), and 2 (Complete). Min=0, Max=2

uint TaskId: Unique ID of the task.

PvsUndefinedDisk

Read-Only Fields

bool VHDX: If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false

string Name: Name of the disk file without the extension.

uint Status: Status of the disk file. Values are: 0 (Valid), 1 (Missing Properties File), 2 (Access Denied), 3 (Access Denied and Missing Properties File), 4 (Invalid Disk File), 5 (Manifest Missing or Invalid), 6 (Both VHD and VHDX)

PvsUpdateTask

Read/Write Fields

uint[] Date: Days of the month. Numbers from 1-31 are the only valid values. This is used with Monthly Date recurrence. Default="" Max Length=83

uint DayMask: Days selected values. 1 = Monday, 2 = Tuesday, 4 = Wednesday, 8 = Thursday, 16 = Friday, 32 = Saturday, 64 = Sunday, 128 = Day.

Default=0. This is used with Weekly and Monthly Type recurrence. Min=1, Max=255, Default=4

string Description: User description. Default="" Max Length=250

string Domain: Domain to add the Disk Update Device(s) to. If not included, the first Domain Controller found on the Server is used. Default="" Max Length=255

bool Enabled: True when it will be processed, false otherwise. Default=true

string EsdType: Esd to use. Valid values are SCCM or WSUS. If no value, a custom script is run on the client. Default="" Max Length=50

uint Hour: The hour of the day to perform the task. Min=0, Max=23, Default=0

uint Minute: The minute of the hour to perform the task. Min=0, Max=59, Default=0

uint MonthlyOffset: When to happen monthly. 0 = None, 1 = First, 2 = Second, 3 = Third, 4 = Forth, 5 = Last. This is used with Monthly Type recurrence. Min=0, Max=5, Default=3

string OrganizationUnit: Organizational Unit to add the Disk Update Device(s) to. This parameter is optional. If it is not specified, the device is added to the built in Computers container. Child OU's should be delimited with forward slashes, e.g. "ParentOU/ChildOU". Special characters in an OU name, such as '"', '#', '+', ',', ';', '>', '=', must be escaped with a backslash. For example, an OU called "commaIn,TheMiddle" must be specified as "commaIn\,TheMiddle". The old syntax of delimiting child OU's with a comma is still supported, but deprecated. Note that in this case, the child OU comes first, e.g. "ChildOU,ParentOU". Default="" Max Length=255

uint PostUpdateApprove: Access to place the version in after the update has occurred. 0 = Production, 1 = Test, 2 = Maintenance. Min=0, Max=2, Default=0

string PostUpdateScript: Script file to run after the update finishes. Default="" Max Length=255

string PostVmScript: Script file to run after the VM is unloaded. Default="" Max Length=255

string PreUpdateScript: Script file to run before the update starts. Default="" Max Length=255

string PreVmScript: Script file to run before the VM is loaded. Default="" Max Length=255

uint Recurrence: The update will reoccur on this schedule. 0 = None, 1 = Daily, 2 = Every Weekday, 3 = Weekly, 4 = Monthly Date, 5 = Monthly Type. Min=0, Max=5, Default=0

string Name or UpdateTaskName: Name of the Update Task. It is unique within the Site. Max Length=50

Read-Only Fields

Guid SiteId: GUID of the Site that this Update Task is a member of. It is not used with SiteName.

string SiteName: Name of the Site that this Update Task is a member of. It is not used with SiteId.

Guid Guid or UpdateTaskId: Read-only GUID that uniquely identifies this Update Task.

PvsVersion

Read-Only Fields

string DbEdition: Edition of the database. If 'Express Edition', monitor dbSize.

uint DbSize: Size of the database in MB. Monitor this value if the edition is 'Express Edition' and this value is close to reaching the 4000 MB maximum. Default=0

uint DbVersion: Version of the database schema as a number. Default=0

string MapiVersion: Version of the system in major.minor.point.build format.

uint MapiVersionNumber: Internal version number of the system. It is a number that is increaed by 100 for each major and minor release. Point releases are the numbers between each 100. Value is 0 when the system does not support MapiVersionNumber. Default=0

string SdkVersion: Version of the SDK in major.minor.build format.

uint Type: Type of system. Values are 0 (Normal), 1 (OROM), and 2 (Secure). Default=0

PvsVirtualHostingPool

Read/Write Fields

string Datacenter: Datacenter of the Virtual Hosting Pool. Default="" Max Length=250

string Description: User description. Default="" Max Length=250

string Password: Password to use when logging into the Server.

string PlatformVersion: Hypervisor Host Version Default="" Max Length=250

uint Port: Port of the Host Server. Min=80, Max=65534, Default=80

bool PrepopulateEnabled: Enable prepopulate when set to true Default=false

string Server: Name or IP of the Host Server. Max Length=255

uint ShutdownTimeout: Timeout for shutdown. Min=2, Max=30, Default=10

uint Type: Type of the Virtual Hosting Pool. 0 = Citrix XenServer, 1 = Microsoft SCVMM/Hyper-V, 2 = VMWare vSphere/ESX. Min=0, Max=3, Default=0

uint UpdateLimit: Number of updates at the same time. Min=2, Max=1000, Default=1000

uint UpdateTimeout: Timeout for updates. Min=2, Max=240, Default=60

string UserName: Name to use when logging into the Server.

string Name or VirtualHostingPoolName: Name of the Virtual Hosting Pool. It is unique within the Site. Max Length=50

string XdHcCustomProperties: Custom Properties for HCL Connection Details object Default="" Max Length=250

string XdHcHypervisorConnectionName: Hypervisor Connection Name for HCL Connection Details object Default="" Max Length=250

string XdHcHypervisorConnectionUid: Hypervisor Connection Uid for HCL Connection Details object Default="" Max Length=250

string XdHcRevision: Revision for HCL Connection Details object Default="" Max Length=250

string XdHcSslThumbprints: Ssl Thumbprints for HCL Connection Details object Default="" Max Length=250

Guid XdHostingUnitUuid: UUID of XenDesktop Hosting Unit Default=00000000-0000-0000-0000-000000000000

Guid XsPvsSiteUuid: UUID of XenServer PVS_site Default=00000000-0000-0000-0000-000000000000

Read-Only Fields

Guid SiteId: GUID of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteName.

string SiteName: Name of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteId.

Guid Guid or VirtualHostingPoolId: Read-only GUID that uniquely identifies this Virtual Hosting Pool.

PvsXDSite

Read/Write Field

string[] ConfigServices: XenDesktop Server addresses. Max Length=2000

Read-Only Field

Guid Guid or XdSiteId: GUID of the XenDesktop Site.

Cmdlets

Add-PvsDeviceToDomain

Add a Device, all Devices in a Collection or View to a Domain.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Add to the Domain.

string[] Name or DeviceName: Name of the Device to Add to the Domain.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Add to the Domain.

Guid[] CollectionId: GUID of the Collection to Add all Devices to the Domain.

Guid[] SiteViewId: GUID of the Site View to Add all Devices to the Domain.

Guid[] FarmViewId: GUID of the Farm View to Add all Devices to the Domain.

string[] FarmViewName: Name of the Farm View to Add all Devices to the Domain.

or one of these required & resolutions

string[] CollectionName: Name of the Collection to Add all Devices to the Domain.

string[] SiteViewName: Name of the Site View to Add all Devices to the Domain.

Optional

string[] Domain: Domain to add the Device(s) to. If not included, the first Domain Controller found on the Server is used.

string[] OrganizationUnit: Organizational Unit to add the Device(s) to. This parameter is optional. If it is not specified, the device is added to the built in Computers container. Child OU's should be delimited with forward slashes, e.g. "ParentOU/ChildOU". Special characters in an OU name, such as '"', '#', '+', ',', ';', '>', '=', must be escaped with a backslash. For example, an OU called "commaIn,TheMiddle" must be specified as "commaIn\,TheMiddle". The old syntax of delimiting child OU's with a comma is still supported, but deprecated. Note that in this case, the child OU comes first, e.g. "ChildOU,ParentOU".

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId, CollectionId, SiteViewId or FarmViewId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Add-PvsDeviceToDomain for Name with Domain and OrganizationUnit

```
Add-PvsDeviceToDomain -Name theDevice -Domain theDomain -
OrganizationUnit theOrganizationUnit
```

EXAMPLE 2: Add-PvsDeviceToDomain for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Add-PvsDeviceToDomain.

```
Get-PvsDevice -Name theDevice -Fields Guid | Add-PvsDeviceToDomain -
Domain theDomain -OrganizationUnit
theOrganizationUnit
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Add-PvsDeviceToDomain for FarmViewName with Domain and OrganizationUnit

```
Add-PvsDeviceToDomain -FarmViewName theFarmView -Domain theDomain -
                OrganizationUnit theOrganizationUnit
```

EXAMPLE 4: Add-PvsDeviceToDomain for PvsFarmView Using Pipe

```
The Get-PvsFarmView output is piped to the Add-PvsDeviceToDomain.
```

```
Get-PvsFarmView -Name theFarmView -Fields Guid | Add-PvsDeviceToDomain
                -Domain theDomain -OrganizationUnit
                theOrganizationUnit
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 5: Add-PvsDeviceToDomain for CollectionName with Domain and OrganizationUnit

```
Add-PvsDeviceToDomain -CollectionName theCollection -SiteName theSite
                -Domain theDomain -OrganizationUnit
                theOrganizationUnit
```

```
CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 6: Add-PvsDeviceToDomain for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Add-PvsDeviceToDomain.
```

```
Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                Add-PvsDeviceToDomain -Domain theDomain -
                OrganizationUnit theOrganizationUnit
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

```
Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

EXAMPLE 7: Add-PvsDeviceToDomain for SiteViewName with Domain and OrganizationUnit

```
Add-PvsDeviceToDomain -SiteViewName theSiteView -SiteName theSite -
                Domain theDomain -OrganizationUnit
                theOrganizationUnit
```

```
SiteViewId can be used instead of SiteViewName so that the SiteName or
                SiteId is not also needed.
```

EXAMPLE 8: Add-PvsDeviceToDomain for PvsSiteView Using Pipe

```
The Get-PvsSiteView output is piped to the Add-PvsDeviceToDomain.
```

```
Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
                Add-PvsDeviceToDomain -Domain theDomain -
                OrganizationUnit theOrganizationUnit
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

```
Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

Add-PvsDeviceToView

Move a Device to a Collection. Personal vDisk Devices cannot be moved to a Collection in another Site.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Add.

string[] Name or DeviceName: Name of the Device to Add.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Add.

One of these required

Guid[] SiteViewId: GUID of the Site View to Add the Device to.

Guid[] FarmViewId: GUID of the Farm View to Add the Device to.

string[] FarmViewName: Name of the Farm View to Add the Device to.

or this required & resolution

string[] SiteViewName: Name of the Site View to Add the Device to.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId, SiteViewId or FarmViewId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Add-PvsDeviceToView for PvsDevice to PvsFarmView

```
Add-PvsDeviceToView -Name theDevice -PvsFarmViewName thePvsFarmView
```

EXAMPLE 2: Add-PvsDeviceToView for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Add-PvsDeviceToView.

```
Get-PvsDevice -Name theDevice -Fields Guid | Add-PvsDeviceToView -
    PvsFarmViewName thePvsFarmView
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Add-PvsDeviceToView for PvsFarmView Using Pipe

The Get-PvsFarmView output is piped to the Add-PvsDeviceToView.

```
Get-PvsFarmView -Name theFarmView -Fields Guid | Add-PvsDeviceToView -
    Name theDevice
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 4: Add-PvsDeviceToView for PvsDevice to PvsSiteView

```
Add-PvsDeviceToView -Name theDevice -SiteViewName theSiteView -
SiteName theSite
```

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

EXAMPLE 5: Add-PvsDeviceToView for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Add-PvsDeviceToView.

```
Get-PvsDevice -Name theDevice -Fields Guid | Add-PvsDeviceToView -
SiteViewName theSiteView -SiteName theSite
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

EXAMPLE 6: Add-PvsDeviceToView for PvsSiteView Using Pipe

The Get-PvsSiteView output is piped to the Add-PvsDeviceToView.

```
Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
Add-PvsDeviceToView -Name theDevice
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Add-PvsDiskLocatorToDevice

Assign a Disk Locator to a Device, a Collection or View.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Assign.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator File to Assign.

One of these required

Guid[] DeviceId: GUID of the Device to Assign a Disk Locator.

string[] DeviceName: Name of the Device to Assign a Disk Locator.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Assign a Disk Locator.

Guid[] CollectionId: GUID of the Collection to Assign a Disk Locator or Locators to all Devices.

Guid[] SiteViewId: GUID of the Site View to Assign a Disk Locator to all Devices.

Guid[] FarmViewId: GUID of the Farm View to Assign a Disk Locator to all Devices.

string[] FarmViewName: Name of the Farm View to Assign a Disk Locator to all Devices.

or one of these required & resolutions

string[] CollectionName: Name of the Collection to Assign a Disk Locator or Locators to all Devices.

string[] SiteViewName: Name of the Site View to Assign a Disk Locator to all Devices.

Optional

SwitchParameter RemoveExisting: If -RemoveExisting is specified, remove the existing Disk Locators before assigning the new one.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId, DeviceId, CollectionId, SiteViewId or FarmViewId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Add-PvsDiskLocatorToDevice for PvsDiskLocator to PvsDevice

```
Add-PvsDiskLocatorToDevice -Name theDiskLocator -DeviceName theDevice
-SiteName theSite -StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 2: Add-PvsDiskLocatorToDevice for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Add-PvsDiskLocatorToDevice.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Fields Guid | Add-PvsDiskLocatorToDevice -
DeviceName theDevice
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 3: Add-PvsDiskLocatorToDevice for PvsDevice Using Pipe

```
The Get-PvsDevice output is piped to the Add-PvsDiskLocatorToDevice.

Get-PvsDevice -Name theDevice -Fields Guid | Add-
                PvsDiskLocatorToDevice -Name theDiskLocator -SiteName
                theSite -StoreName theStoreName

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 4: Add-PvsDiskLocatorToDevice for PvsDiskLocator to PvsCollection

```
Add-PvsDiskLocatorToDevice -Name theDiskLocator -CollectionName
                theCollection -SiteName theSite -StoreName theStore
```

EXAMPLE 5: Add-PvsDiskLocatorToDevice for PvsDiskLocator Using Pipe

```
The Get-PvsDiskLocator output is piped to the Add-
                PvsDiskLocatorToDevice.

Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Add-PvsDiskLocatorToDevice -
                CollectionName theCollection -SiteName theSite

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 6: Add-PvsDiskLocatorToDevice for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Add-
                PvsDiskLocatorToDevice.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                Add-PvsDiskLocatorToDevice -Name theDiskLocator -
                SiteName theSite -StoreName theStoreName

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 7: Add-PvsDiskLocatorToDevice for PvsDiskLocator to PvsFarmView

```
Add-PvsDiskLocatorToDevice -Name theDiskLocator -FarmViewName
                theFarmView -SiteName theSite -StoreName theStore

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 8: Add-PvsDiskLocatorToDevice for PvsDiskLocator Using Pipe

```
The Get-PvsDiskLocator output is piped to the Add-
                PvsDiskLocatorToDevice.
```

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Add-PvsDiskLocatorToDevice -
                FarmViewName theFarmView

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 9: Add-PvsDiskLocatorToDevice for PvsFarmView Using Pipe

```
The Get-PvsFarmView output is piped to the Add-PvsDiskLocatorToDevice.

Get-PvsFarmView -Name theFarmView -Fields Guid | Add-
                PvsDiskLocatorToDevice -Name theDiskLocator -SiteName
                theSite -StoreName theStoreName

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 10: Add-PvsDiskLocatorToDevice for PvsDiskLocator to PvsSiteView

```
Add-PvsDiskLocatorToDevice -Name theDiskLocator -SiteViewName
                theSiteView -SiteName theSite -StoreName theStore
```

EXAMPLE 11: Add-PvsDiskLocatorToDevice for PvsDiskLocator Using Pipe

```
The Get-PvsDiskLocator output is piped to the Add-
                PvsDiskLocatorToDevice.

Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Add-PvsDiskLocatorToDevice -
                SiteViewName theSiteView -SiteName theSite

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 12: Add-PvsDiskLocatorToDevice for PvsSiteView Using Pipe

```
The Get-PvsSiteView output is piped to the Add-PvsDiskLocatorToDevice.

Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
                Add-PvsDiskLocatorToDevice -Name theDiskLocator -
                SiteName theSite -StoreName theStoreName

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

Add-PvsDiskToUpdateTask

```
Add a Disk to an Update Task.

    This required

     Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Assign.

    or this required & resolution
```

string[] Name or DiskLocatorName: Name of the Disk Locator File to Assign.

One of these required

Guid[] UpdateTaskId: GUID of the Update Task to Assign a Disk.

string[] UpdateTaskName: Name of the Update Task to Assign a Disk.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId or UpdateTaskId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Add-PvsDiskToUpdateTask for PvsDiskLocator to PvsUpdateTask

```
Add-PvsDiskToUpdateTask -Name theDiskLocator -UpdateTaskName
theUpdateTask -SiteName theSite -StoreName theStore
```

UpdateTaskId can be used instead of UpdateTaskName so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Add-PvsDiskToUpdateTask for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Add-PvsDiskToUpdateTask.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Fields Guid | Add-PvsDiskToUpdateTask -
UpdateTaskName theUpdateTask -SiteName theSite
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Add-PvsDiskToUpdateTask for PvsUpdateTask Using Pipe

The Get-PvsUpdateTask output is piped to the Add-PvsDiskToUpdateTask.

```
Get-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Fields Guid |
Add-PvsDiskToUpdateTask -Name theDiskLocator -
SiteName theSite -StoreName theStore
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Add-PvsDiskVersion

Add one or more new Versions to a Disk. A manifest file for the new Disk Version(s) must exist in the Store.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator File to Add the new Disk Version(s) to.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator File to Add the new Disk Version(s) to.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Add-PvsDiskVersion for Name

```
Add-PvsDiskVersion -Name theDiskLocator -SiteName theSite -StoreName
theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 2: Add-PvsDiskVersion for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Add-PvsDiskVersion.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Fields Guid | Add-PvsDiskVersion
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Clear-PvsConnection

Closes the existing SoapServer connection, and if -Persist is specified the connection settings in the registry are removed.

Optional

SwitchParameter Persist: If -Persist is specified, clear the connection settings in the registry.

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Clear-PvsConnection

```
Clear-PvsConnection
```

EXAMPLE 2: Clear-PvsConnection with Persist

```
Clear-PvsConnection -Persist
```

Clear-PvsTask

Clear a single or all completed or cancelled Tasks in a Site or the whole Farm.

One of these optional

uint TaskId: Id of the Task to Clear.

Guid[] SiteId: Site Id of the Tasks to Clear.

string[] SiteName: Site Name of the Tasks to Clear.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

TaskId or SiteId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Clear-PvsTask for TaskId

```
Clear-PvsTask -TaskId 101
```

EXAMPLE 2: Clear-PvsTask for PvsTask Using Pipe

The Get-PvsTask output is piped to the Clear-PvsTask.

```
Get-PvsTask -TaskId 101 -Fields -TaskId | Clear-PvsTask
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Clear-PvsTask for SiteName

```
Clear-PvsTask -SiteName theSite
```

EXAMPLE 4: Clear-PvsTask for PvsTask Using Pipe

```
The Get-PvsTask output is piped to the Clear-PvsTask.

Get-PvsTask -SiteName theSite -Fields -SiteId | Clear-PvsTask

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

Copy-PvsDeviceProperties

```
Copy properties of one Device to a Device, all the Devices in a Collection,
  Site View or Farm View.

   One of these required

    Guid DeviceIdFrom: GUID of the Device to Copy from.

    string DeviceNameFrom: Name of the Device to Copy from.

    PvsPhysicalAddress DeviceMacFrom: Mac of the Device to Copy from.

   One of these required

    Guid Guid or DeviceId: GUID of the Device to Copy to.

    string Name or DeviceName: Name of the Device to Copy to.

    PvsPhysicalAddress DeviceMac: MAC of the Device to Copy to.

    Guid CollectionId: GUID of the Collection to Copy to.

    Guid SiteViewId: GUID of the Site View to Copy to.

    Guid FarmViewId: GUID of the Farm View to Copy to.

    string FarmViewName: Name of the Farm View to Copy to.

   or one of these required & resolutions

    string CollectionName: Name of the Collection to Copy to.

    string SiteViewName: Name of the Site View to Copy to.

   Optional

    uint[] Properties: If not specified, all are copied. Properties to
                copy. Values are: 1 (Description), 2 (Class), 3
                (Port), 4 (Disabled), 5 (Boot Behavior), 6 (Disk
                Assignment), 7 (Personality), 8 (Printer), 9 (Type),
                10 (Authentication) and 11 (Logging).

   One of these resolutions when needed

    Guid SiteId: GUID of the Site.

    string SiteName: Name of the Site.

   Instead of a parameter that matches one of the members listed

    PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

     DeviceId, CollectionId, SiteViewId or FarmViewId
```

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Copy-PvsDeviceProperties for Name

```
Copy-PvsDeviceProperties -DeviceNameFrom theDeviceFrom -Name
theDeviceTo
```

EXAMPLE 2: Copy-PvsDeviceProperties for Name with Properties

```
Copy-PvsDeviceProperties -DeviceNameFrom theDeviceFrom -Name
theDeviceTo -Properties 2, 3
```

EXAMPLE 3: Copy-PvsDeviceProperties for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Copy-PvsDeviceProperties.

```
Get-PvsDevice -Name theDeviceTo -Fields Guid | Copy-
PvsDeviceProperties -DeviceNameFrom theDeviceFrom -
Properties 2, 3
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 4: Copy-PvsDeviceProperties for Name

```
Copy-PvsDeviceProperties -DeviceMacFrom "00-11-22-33-44-55" -Name
theDeviceTo
```

EXAMPLE 5: Copy-PvsDeviceProperties for Name with Properties

```
Copy-PvsDeviceProperties -DeviceMacFrom "00-11-22-33-44-55" -Name
theDeviceTo -Properties 2, 3
```

EXAMPLE 6: Copy-PvsDeviceProperties for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Copy-PvsDeviceProperties.

```
Get-PvsDevice -Name theDeviceTo -Fields Guid | Copy-
PvsDeviceProperties -DeviceMacFrom "00-11-22-33-44-
55" -Properties 2, 3
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 7: Copy-PvsDeviceProperties for FarmViewName

```
Copy-PvsDeviceProperties -DeviceNameFrom theDeviceFrom -FarmViewName
theFarmViewNameTo
```

EXAMPLE 8: Copy-PvsDeviceProperties for FarmViewName with Properties

```
Copy-PvsDeviceProperties -DeviceNameFrom theDeviceFrom -FarmViewName
theFarmViewNameTo -Properties 2, 3
```

EXAMPLE 9: Copy-PvsDeviceProperties for PvsFarmView Using Pipe

The Get-PvsFarmView output is piped to the Copy-PvsDeviceProperties.

```
Get-PvsFarmView -Name theFarmViewTo -Fields Guid | Copy-
                PvsDeviceProperties -DeviceNameFrom theDeviceFrom -
                Properties 2, 3
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 10: Copy-PvsDeviceProperties for CollectionName

```
Copy-PvsDeviceProperties -DeviceNameFrom theDeviceFrom -CollectionName
                theCollectionNameTo -SiteName theSite
```

```
CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 11: Copy-PvsDeviceProperties for CollectionName with Properties

```
Copy-PvsDeviceProperties -DeviceNameFrom theDeviceFrom -CollectionName
                theCollectionNameTo -SiteName theSite -Properties 2,
                3
```

```
CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 12: Copy-PvsDeviceProperties for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Copy-PvsDeviceProperties.
```

```
Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                Copy-PvsDeviceProperties -DeviceNameFrom
                theDeviceFrom -Properties 2, 3
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

```
Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

EXAMPLE 13: Copy-PvsDeviceProperties for SiteViewName

```
Copy-PvsDeviceProperties -DeviceNameFrom theDeviceFrom -SiteViewName
                theSiteViewNameTo -SiteName theSite
```

```
SiteViewId can be used instead of SiteViewName so that the SiteName or
                SiteId is not also needed.
```

EXAMPLE 14: Copy-PvsDeviceProperties for SiteViewName with Properties

```
Copy-PvsDeviceProperties -DeviceNameFrom theDeviceFrom -SiteViewName
                theSiteViewNameTo -SiteName theSite -Properties 2, 3
```

```
SiteViewId can be used instead of SiteViewName so that the SiteName or
                SiteId is not also needed.
```

EXAMPLE 15: Copy-PvsDeviceProperties for PvsSiteView Using Pipe

```
The Get-PvsSiteView output is piped to the Copy-PvsDeviceProperties.
```

```
Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
                Copy-PvsDeviceProperties -DeviceNameFrom
                theDeviceFrom -Properties 2, 3
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Copy-PvsDiskProperties

Copy properties of one Disk to a Disk.

This required

Guid DiskLocatorIdFrom: GUID of the Disk Locator to Copy from.

This required

Guid Guid or DiskLocatorId: GUID of the Disk Locator to Copy to.

Optional

uint[] Properties: If not specified, all are copied. Properties to copy. Values are: 1 (Description), 2 (Class), 3 (Type), 4 (Disk Mode), 5 (Auto Update), 6 (HA), 7 (Active Directory), 8 (Printer), 10 (Version), 11 (Date), 12 (Author), 13 (Title), 14 (Company), 15 (Internal Filename), 16 (Original Filename), 17 (Hardware Target), 18 (Menu Text), 19 (Enabled), 20 (Server), and 21 (Store).

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Copy-PvsDiskProperties for Guid

```
Copy-PvsDiskProperties -DiskLocatorIdFrom "66302103-2991-4e42-ba58-
a1614cec070c" -Guid "f5eb3de9-bcf4-416f-a289-
6a9472c13f8b"
```

EXAMPLE 2: Copy-PvsDiskProperties for Guid with Properties

```
Copy-PvsDiskProperties -DiskLocatorIdFrom "66302103-2991-4e42-ba58-
a1614cec070c" -Guid "f5eb3de9-bcf4-416f-a289-
6a9472c13f8b" -Properties 2, 3
```

EXAMPLE 3: Copy-PvsDiskProperties for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Copy-PvsDiskProperties.

```
Get-PvsDiskLocator -Guid "f5eb3de9-bcf4-416f-a289-6a9472c13f8b" -
Fields Guid | Copy-PvsDiskProperties -
DiskLocatorIdFrom "66302103-2991-4e42-ba58-
a1614cec070c" -Properties 2, 3
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Copy-PvsServerProperties

Copy properties of one Server to a Server.

One of these required

Guid ServerIdFrom: GUID of the Server to Copy from.

string ServerNameFrom: Name of the Server to Copy from.

One of these required

Guid Guid or ServerId: GUID of the Server to Copy to.

string Name or ServerName: Name of the Server to Copy to.

Optional

uint[] Properties: If not specified, all are copied. Properties to copy. Values are: 1 (Configuration), 2 (Port), 4 (Active Directory), 5 (Advanced Server), 6 (Advanced Network), 7 (Advanced Pacing), 8 (Advanced Device) and 9 (Logging).

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Copy-PvsDiskProperties for Name

```
Copy-PvsDiskProperties -ServerNameFrom theServerFrom -Name theServerTo
```

EXAMPLE 2: Copy-PvsDiskProperties for Name with Properties

```
Copy-PvsDiskProperties -ServerNameFrom theServerFrom -Name theServerTo
-Properties 2, 3
```

EXAMPLE 3: Copy-PvsDiskProperties for PvsServer Using Pipe

The Get-PvsServer output is piped to the Copy-PvsDiskProperties.

```
Get-PvsServer -Name theServerTo -Fields Guid | Copy-PvsDiskProperties
-ServerNameFrom theServerFrom -Properties 2, 3
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Disable-PvsDeviceDiskLocator

Disable a Device's DiskLocator.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Disable the DiskLocator for.

string[] Name or DeviceName: Name of the Device to Disable the DiskLocator for.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Disable the DiskLocator for.

This required

Guid[] DiskLocatorId: GUID of the DiskLocator to Disable for the Device.

or this required & resolution

string[] DiskLocatorName: Name of the DiskLocator File to Disable for the Device.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId or DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Disable a PvsDevice PvsDiskLocator

This example disables the PvsDiskLocator named theDiskLocator for the PvsDevice named theDevice.

```
Disable-PvsDeviceDiskLocator -Name theDevice -DiskLocatorName
theDiskLocator -StoreName theStore
```

DiskLocatorId can be used instead of DiskLocatorName so that the StoreName or StoreId are not also needed.

Dismount-PvsDisk

No longer Map the Disk.

EXAMPLE 1: Dismount-PvsDisk

```
Dismount-PvsDisk
```

Enable-PvsDeviceDiskLocator

Enable a Device's DiskLocator. If the DiskLocator is Disabled, that overrides the Device DiskLocator setting.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Enable the DiskLocator for.

string[] Name or DeviceName: Name of the Device to Enable the DiskLocator for.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Enable the DiskLocator for.

This required

Guid[] DiskLocatorId: GUID of the DiskLocator to Enable for the Device.

or this required & resolution

string[] DiskLocatorName: Name of the DiskLocator to Enable for the Device.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId or DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Enable a PvsDevice PvsDiskLocator

This example enables the PvsDiskLocator named theDiskLocator for the PvsDevice named theDevice.

Enable-PvsDeviceDiskLocator -Name theDevice -DiskLocatorName theDiskLocator -StoreName theStore

DiskLocatorId can be used instead of DiskLocatorName so that the StoreName or StoreId are not also needed.

Export-PvsAuditTrail

Archive the information in the Audit Trail up to a certain date to a file. When finished, the information archived will be removed from the Audit Trail.

This required

string[] FileName: Name of the file to archive the Audit Trail to.
This must be a full file path name.

Optional

DateTime EndDate: Last date of information to Archive. If not entered,
all information is Archived. Uses only the date.

SwitchParameter NoPurgeData: If -NoPurgeData is specified, the
information archived will not be removed from the
Audit Trail.

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -
Confirm is specified, the operation will be
confirmed. $ConfirmPreference can be set to "low" to
have confirmation without the Confirm parameter.

EXAMPLE 1: Export-AuditTrail for PvsFarm

```
Export-AuditTrail -Name "C:\export\theFileName"
```

EXAMPLE 2: Export-AuditTrail for PvsFarm with EndDate

```
Export-AuditTrail -EndDate  "01/01/2015" -Name "C:\export\theFileName"
```

EXAMPLE 3: Export-AuditTrail for PvsFarm with EndDate and NoPurgeData

```
Export-AuditTrail -EndDate  "01/01/2015" -NoPurgeData -Name
"C:\export\theFileName"
```

Export-PvsDisk

Export the disk stack to a manifest file.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator that identifies
the disk to export.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator that
identifies the disk to export.

Optional

uint Version: Version to use as the start of the export. The export
will include all versions starting with this to the
highest one.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a
DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a
DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

EXAMPLE 1: Export-PvsDisk for Name

Export-PvsDisk -Name theDiskLocator -SiteName theSite -StoreName theStore

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 2: Export-PvsDisk for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Export-PvsDisk.

Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName theStore -Fields Guid | Export-PvsDisk

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 3: Export-PvsDisk for Name with Version

Export-PvsDisk -Name theDiskLocator -SiteName theSite -StoreName theStore -Version 4

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 4: Export-PvsDisk for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Export-PvsDisk.

Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName theStore -Fields Guid | Export-PvsDisk -Version 4

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Export-PvsOemLicenses

Oem Only: Export the Oem Licenses for the Devices to the fileName specified.

This required

string FileName: Name of the file to export the Oem Licenses to. This must be a full file path name.

One of these required

Guid Guid or DeviceId: GUID of the Device to Export Oem Licenses to the fileName specified.

string Name or DeviceName: Name of the Device to Export Oem Licenses to the fileName specified.

PvsPhysicalAddress DeviceMac: MAC of the Device to Export Oem Licenses to the fileName specified.

Guid CollectionId: GUID of the Collection to Export all Device Oem Licenses to the fileName specified.

Guid SiteViewId: GUID of the Site View to Export all Device Oem Licenses to the fileName specified.

Guid FarmViewId: GUID of the Farm View to Export all Device Oem Licenses to the fileName specified.

string FarmViewName: Name of the Farm View to Export all Device Oem Licenses to the fileName specified.

or one of these required & resolutions

string CollectionName: Name of the Collection to Export all Device Oem Licenses to the fileName specified.

string SiteViewName: Name of the Site View to Export all Device Oem Licenses to the fileName specified.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId, CollectionId, SiteViewId or FarmViewId

EXAMPLE 1: Export-PvsOemLicenses for Name

```
Export-PvsOemLicenses -Name theDevice -FileName
"C:\export\theFileName"
```

EXAMPLE 2: Export-PvsOemLicenses for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Export-PvsOemLicenses.

```
Get-PvsDevice -Name theDevice -Fields Guid | Export-PvsOemLicenses -
FileName "C:\export\theFileName"
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Export-PvsOemLicenses for DeviceMac

```
Export-PvsOemLicenses -DeviceMac "00-11-22-33-44-55" -FileName
"C:\export\theFileName"
```

EXAMPLE 4: Export-PvsOemLicenses for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Export-PvsOemLicenses.

```
Get-PvsDevice -DeviceMac "00-11-22-33-44-55" -Fields Guid | Export-
PvsOemLicenses -FileName "C:\export\theFileName"
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 5: Export-PvsOemLicenses for FarmViewName

```
Export-PvsOemLicenses -FarmViewName theFarmView -FileName
                "C:\export\theFileName"
```

EXAMPLE 6: Export-PvsOemLicenses for PvsFarmView Using Pipe

```
The Get-PvsFarmView output is piped to the Export-PvsOemLicenses.

Get-PvsFarmView -Name theFarmView -Fields Guid | Export-PvsOemLicenses
                -FileName "C:\export\theFileName"

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 7: Export-PvsOemLicenses for CollectionName

```
Export-PvsOemLicenses -CollectionName theCollection -SiteName theSite
                -FileName "C:\export\theFileName"

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 8: Export-PvsOemLicenses for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Export-PvsOemLicenses.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                Export-PvsOemLicenses -FileName
                "C:\export\theFileName"

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

EXAMPLE 9: Export-PvsOemLicenses for SiteViewName

```
Export-PvsOemLicenses -SiteViewName theSiteView -SiteName theSite -
                FileName "C:\export\theFileName"

SiteViewId can be used instead of SiteViewName so that the SiteName or
                SiteId is not also needed.
```

EXAMPLE 10: Export-PvsOemLicenses for PvsSiteView Using Pipe

```
The Get-PvsSiteView output is piped to the Export-PvsOemLicenses.

Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
                Export-PvsOemLicenses -FileName
                "C:\export\theFileName"

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

Get-PvsADAccount

Return a PvsAdAccount object if the named Device Account in the domain is found.

This required

string Domain: Domain the account is a member of.

This required

string Name: Name of the Device for the account.

PvsADAccount: If successful, the PvsADAccount object is returned.

EXAMPLE 1: Get PvsADAccount

Get the PvsADAccount in the Domain named theDomain for the Device named theDevice.

```
Get-PvsADAccount -Domain theDomain -Name theDevice
```

Get-PvsAuditActionParameter

Get the Parameters of an Audit Action.

This required

Guid[] AuditActionId: GUID of the Audit Action to Get Parameters for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

AuditActionId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name or AuditParameterName: Name of the parameter. Max Length=50

Value: Value of the parameter. Max Length=1000

PvsAuditActionParameter[]: If successful, the PvsAuditActionParameter object(s) are returned.

EXAMPLE 1: Get PvsAuditActionParameter

Get all PvsAuditActionParameter for the AuditActionId e8baa554-7c2d-49e5-9f6b-e0bc46179fc7.

```
Get-PvsAuditActionParameter -AuditActionId "e8baa554-7c2d-49e5-9f6b-e0bc46179fc7"
```

EXAMPLE 2: Get PvsAuditActionParameter for Multiple AuditActionId

Get all PvsAuditActionParameter for the AuditActionId e8baa554-7c2d-49e5-9f6b-e0bc46179fc7 and 54ee6180-7fbc-42a2-9499-2e4936f039dc.

```
Get-PvsAuditActionParameter -AuditActionId "e8baa554-7c2d-49e5-9f6b-e0bc46179fc7", "54ee6180-7fbc-42a2-9499-2e4936f039dc"
```

EXAMPLE 3: Get PvsAuditActionParameter for Get-PvsAuditTrail Results with Parameters

```
Get-PvsAuditTrail is called and only the PvsAuditTrail.Attachment with
                bit 4 (Parameters) set are used to call Get-
                PvsAuditActionParameter.

Get-PvsAuditTrail | Where-Object {($_.Attachments -band 4) -eq 4} |
                Get-PvsAuditActionParameter
```

Get-PvsAuditActionProperty

```
Get the Properties of an Audit Action.

    This required

     Guid[] AuditActionId: GUID of the Audit Action to Get Properties for.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      AuditActionId

    If only selected fields are needed, pass them in the Fields parameter
      as a string array.

     Name or AuditPropertyName: Name of the property. Max Length=50

     OldValue: Previous value of the Property. Default="" Max Length=1000

     NewValue: New value of the Property. Default="" Max Length=1000

    PvsAuditActionProperty[]: If successful, the PvsAuditActionProperty
      object(s) are returned.
```

EXAMPLE 1: Get PvsAuditActionProperty

```
Get all PvsAuditActionProperty for the AuditActionId e8baa554-7c2d-
                49e5-9f6b-e0bc46179fc7.

Get-PvsAuditActionProperty -AuditActionId "e8baa554-7c2d-49e5-9f6b-
                e0bc46179fc7"
```

EXAMPLE 2: Get PvsAuditActionProperty for Multiple AuditActionId

```
Get all PvsAuditActionProperty for the AuditActionId e8baa554-7c2d-
                49e5-9f6b-e0bc46179fc7 and 54ee6180-7fbc-42a2-9499-
                2e4936f039dc.

Get-PvsAuditActionProperty -AuditActionId "e8baa554-7c2d-49e5-9f6b-
                e0bc46179fc7", "54ee6180-7fbc-42a2-9499-2e4936f039dc"
```

EXAMPLE 3: Get PvsAuditActionProperty for Get-PvsAuditTrail Results with Properties

```
Get-PvsAuditTrail is called and only the PvsAuditTrail.Attachment with
                bit 8 (Properties) set are used to call Get-
                PvsAuditActionProperty.

Get-PvsAuditTrail | Where-Object {($_.Attachments -band 8) -eq 8} |
                Get-PvsAuditActionProperty
```

Get-PvsAuditActionSibling

Get the Sibling of an Audit Action. It is the 2nd object involved with the action.

This required

Guid[] Guid or AuditActionId: GUID of the Audit Action to Get Sibling for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

AuditActionId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or AuditActionId: GUID of the action.

Type: Type of object that action was performed on. Values are: 1 (AuthGroup), 2 (Collection), 3 (Device), 4 (Disk), 5 (DiskLocator), 6 (Farm), 7 (FarmView), 8 (Server), 9 (Site), 10 (SiteView), 11 (Store), 12 (System), and 13 (UserGroup)

ObjectId: GUID of the object of the action.

ObjectName: Name of the object of the action. Max Length=1000

Path: Path of the object of the action. An example is Site\Collection for a Device. Default="" Max Length=101

SiteId: GUID of the Site for the object of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

SubId: GUID of the Collection or Store of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

PvsAuditAction[]: If successful, the PvsAuditAction object(s) are returned.

EXAMPLE 1: Get PvsAuditActionSibling

Get all PvsAuditActionSibling for the AuditActionId e8baa554-7c2d-49e5-9f6b-e0bc46179fc7.

```
Get-PvsAuditActionSibling -Guid "e8baa554-7c2d-49e5-9f6b-e0bc46179fc7"
```

EXAMPLE 2: Get PvsAuditActionSibling for Multiple AuditActionId

Get all PvsAuditActionSibling for the AuditActionId e8baa554-7c2d-49e5-9f6b-e0bc46179fc7 and 54ee6180-7fbc-42a2-9499-2e4936f039dc.

```
Get-PvsAuditActionSibling -Guid "e8baa554-7c2d-49e5-9f6b-
e0bc46179fc7", "54ee6180-7fbc-42a2-9499-2e4936f039dc"
```

EXAMPLE 3: Get PvsAuditActionSibling for Get-PvsAuditTrail Results with Siblings

```
Get-PvsAuditTrail is called and only the PvsAuditTrail.Attachment with
                bit 2 (Siblings) set are used to call Get-
                PvsAuditActionSibling.

Get-PvsAuditTrail | Where-Object {($_.Attachments -band 2) -eq 2} |
                Get-PvsAuditActionSibling
```

Get-PvsAuditTrail

```
Get the Audit Trail actions for a Farm, Site, Server, DiskLocator,
   Collection, Device, User Group, Site View, Farm View or Store. All Audit
   Trail actions are returned if no parameters are passed. The result can be
   filtered by parent, user\domain and date range.

    One of these optional

     Guid[] Guid or AuditActionId: GUID of the Audit Action to Get.

     Guid[] ParentId: Parent AuditActionId of the records to retrieve. If
                no parameters are included, only records with no
                parent are returned.

     Guid[] RootId: Root AuditActionId of the records to retrieve. All of
                the actions caused by the root action are returned.
                If no parameters are included, only records with no
                root are returned.

     Guid[] SiteId: GUID of the Site to get the Audit Trail for.

     string[] SiteName: Name of the Site to get the Audit Trail for.

     Guid[] CollectionId: GUID of the Collection to get the Audit Trail
                for.

     Guid[] SiteViewId: GUID of the Site View to get the Audit Trail for.

     Guid[] FarmViewId: GUID of the Farm View to get the Audit Trail for.

     string[] FarmViewName: Name of the Farm View to get the Audit Trail
                for.

     Guid[] ServerId: GUID of the Server to get the Audit Trail for.

     string[] ServerName: Name of the Server to get the Audit Trail for.

     Guid[] DeviceId: GUID of the Device to get the Audit Trail for.

     string[] DeviceName: Name of the Device to get the Audit Trail for.

     PvsPhysicalAddress[] DeviceMac: MAC of the Device to get the Audit
                Trail for.

     Guid[] StoreId: GUID of the Store to get the Audit Trail for.

     string[] StoreName: Name of the Store to get the Audit Trail for.

     Guid[] DiskLocatorId: GUID of the DiskLocator to get the Audit Trail
                for.

    or one of these optional & resolutions

     string[] CollectionName: Name of the Collection to get the Audit Trail
                for.
```

string[] SiteViewName: Name of the Site View to get the Audit Trail for.

string[] DiskLocatorName: Name of the DiskLocator to get the Audit Trail for.

Optional

string[] UserName: User that performed the action.

string[] Domain: Domain of the user that performed the action.

DateTime BeginDate: Date/Time of the first actions to get. If not included, all actions until the endDate are returned. If neither this or the endDate are included, then only actions that occurred in the last week are returned. Uses only the date, hour and minute. If connected to a Server version previous to 7.7, only the date is used.

DateTime EndDate: Date/Time of the last actions to get. If not included, all actions from the beginDate until now are returned. If neither this or the beginDate are included, then only actions that occurred in the last week are returned. Uses only the date, hour and minute. If connected to a Server version previous to 7.7, only the date is used.

uint[] Type: Types to get. Values are: 0 (Many), 1 (AuthGroup), 2 (Collection), 3 (Device), 4 (Disk), 5 (DiskLocator), 6 (Farm), 7 (FarmView), 8 (Server), 9 (Site), 10 (SiteView), 11 (Store), 12 (System), and 13 (UserGroup)

uint[] Action: Actions to get. Values are: 1 (AddAuthGroup), 2 (AddCollection), 3 (AddDevice), 4 (AddDiskLocator), 5 (AddFarmView), 6 (AddServer), 7 (AddSite), 8 (AddSiteView), 9 (AddStore), 10 (AddUserGroup), 11 (AddVirtualHostingPool), 12 (AddUpdateTask), 13 (AddDiskUpdateDevice), 1001 (DeleteAuthGroup), 1002 (DeleteCollection), 1003 (DeleteDevice), 1004 (DeleteDeviceDiskCacheFile), 1005 (DeleteDiskLocator), 1006 (DeleteFarmView), 1007 (DeleteServer), 1008 (DeleteServerStore), 1009 (DeleteSite), 1010 (DeleteSiteView), 1011 (DeleteStore), 1012 (DeleteUserGroup), 1013 (DeleteVirtualHostingPool), 1014 (DeleteUpdateTask), 1015 (DeleteDiskUpdateDevice), 1016 (DeleteDiskVersion), 2001 (RunAddDeviceToDomain), 2002 (RunApplyAutoUpdate), 2003 (RunApplyIncrementalUpdate), 2004 (RunArchiveAuditTrail), 2005 (RunAssignAuthGroup), 2006 (RunAssignDevice), 2007 (RunAssignDiskLocator), 2008 (RunAssignServer), 2009 (RunWithReturnBoot), 2010 (RunCopyPasteDevice), 2011 (RunCopyPasteDisk), 2012 (RunCopyPasteServer), 2013 (RunCreateDirectory), 2014 (RunCreateDiskCancel), 2015 (RunDisableCollection), 2016 (RunDisableDevice), 2017 (RunDisableDeviceDiskLocator), 2018 (RunDisableDiskLocator), 2019 (RunDisableUserGroup),

2020 (RunDisableUserGroupDiskLocator), 2021 (RunWithReturnDisplayMessage), 2022 (RunEnableCollection), 2023 (RunEnableDevice), 2024 (RunEnableDeviceDiskLocator), 2025 (RunEnableDiskLocator), 2026 (RunEnableUserGroup), 2027 (RunEnableUserGroupDiskLocator), 2028 (RunExportOemLicenses), 2029 (RunImportDatabase), 2030 (RunImportDevices), 2031 (RunImportOemLicenses), 2032 (RunMarkDown), 2033 (RunWithReturnReboot), 2034 (RunRemoveAuthGroup), 2035 (RunRemoveDevice), 2036 (RunRemoveDeviceFromDomain), 2037 (RunRemoveDirectory), 2038 (RunRemoveDiskLocator), 2039 (RunResetDeviceForDomain), 2040 (RunResetDatabaseConnection), 2041 (RunRestartStreamingService), 2042 (RunWithReturnShutdown), 2043 (RunStartStreamingService), 2044 (RunStopStreamingService), 2045 (RunUnlockAllDisk), 2046 (RunUnlockDisk), 2047 (RunServerStoreVolumeAccess), 2048 (RunServerStoreVolumeMode), 2049 (RunMergeDisk), 2050 (RunRevertDiskVersion), 2051 (RunPromoteDiskVersion), 2052 (RunCancelDiskMaintenance), 2053 (RunActivateDevice), 2054 (RunAddDiskVersion), 2055 (RunExportDisk), 2056 (RunAssignDisk), 2057 (RunRemoveDisk), 2058 (RunDiskUpdateStart), 2059 (RunDiskUpdateCancel), 2060 (RunSetOverrideVersion), 2061 (RunCancelTask), 2062 (RunClearTask), 2063 (RunForceInventory), 2064 RunUpdateBDM, 2065 (RunStartDeviceDiskTempVersionMode), 2066 (RunStopDeviceDiskTempVersionMode), 3001 (RunWithReturnCreateDisk), 3002 (RunWithReturnCreateDiskStatus), 3003 (RunWithReturnMapDisk), 3004 (RunWithReturnRebalanceDevices), 3005 (RunWithReturnCreateMaintenanceVersion), 3006 (RunWithReturnImportDisk), 4001 (RunByteArrayInputImportDevices), 4002 (RunByteArrayInputImportOemLicenses), 5001 (RunByteArrayOutputArchiveAuditTrail), 5002 (RunByteArrayOutputExportOemLicenses), 6001 (SetAuthGroup), 6002 (SetCollection), 6003 (SetDevice), 6004 (SetDisk), 6005 (SetDiskLocator), 6006 (SetFarm), 6007 (SetFarmView), 6008 (SetServer), 6009 (SetServerBiosBootstrap), 6010 (SetServerBootstrap), 6011 (SetServerStore), 6012 (SetSite), 6013 (SetSiteView), 6014 (SetStore), 6015 (SetUserGroup), 6016 SetVirtualHostingPool, 6017 SetUpdateTask, 6018 SetDiskUpdateDevice, 7001 (SetListDeviceBootstraps), 7002 (SetListDeviceBootstrapsDelete), 7003 (SetListDeviceBootstrapsAdd), 7004 (SetListDeviceCustomProperty), 7005 (SetListDeviceCustomPropertyDelete), 7006 (SetListDeviceCustomPropertyAdd), 7007 (SetListDeviceDiskPrinters), 7008 (SetListDeviceDiskPrintersDelete), 7009

(SetListDeviceDiskPrintersAdd), 7010
(SetListDevicePersonality), 7011
(SetListDevicePersonalityDelete), 7012
(SetListDevicePersonalityAdd), 7013
(SetListDiskLocatorCustomProperty), 7014
(SetListDiskLocatorCustomPropertyDelete), 7015
(SetListDiskLocatorCustomPropertyAdd), 7016
(SetListServerCustomProperty), 7017
(SetListServerCustomPropertyDelete), 7018
(SetListServerCustomPropertyAdd), 7019
(SetListUserGroupCustomProperty), 7020
(SetListUserGroupCustomPropertyDelete), and 7021
(SetListUserGroupCustomPropertyAdd)

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to get the Audit Trail for.

string[] SiteName: Name of the Site to get the Audit Trail for.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store to get the Audit Trail for.

string[] StoreName: Name of the Store to get the Audit Trail for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

AuditActionId, ParentId, RootId, SiteId, CollectionId, SiteViewId, FarmViewId, ServerId, DeviceId, StoreId or DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or AuditActionId: GUID of the action.

Time: Date/Time the action occurred down to the millisecond. Has the date and time including milliseconds. Default=Empty

UserName: User that performed the action. Max Length=255

Domain: Domain of the user that performed the action. Max Length=255

Type: Type of object that action was performed on. Values are: 0 (Many), 1 (AuthGroup), 2 (Collection), 3 (Device), 4 (Disk), 5 (DiskLocator), 6 (Farm), 7 (FarmView), 8 (Server), 9 (Site), 10 (SiteView), 11 (Store), 12 (System), and 13 (UserGroup)

Action: Name of the action taken. This is a number that is converted to a string for display. Values are: 1 (AddAuthGroup), 2 (AddCollection), 3 (AddDevice), 4 (AddDiskLocator), 5 (AddFarmView), 6 (AddServer), 7 (AddSite), 8 (AddSiteView), 9 (AddStore), 10 (AddUserGroup), 11 (AddVirtualHostingPool), 12 (AddUpdateTask), 13 (AddDiskUpdateDevice), 1001 (DeleteAuthGroup), 1002 (DeleteCollection), 1003 (DeleteDevice), 1004 (DeleteDeviceDiskCacheFile), 1005 (DeleteDiskLocator), 1006 (DeleteFarmView), 1007 (DeleteServer), 1008 (DeleteServerStore), 1009

(DeleteSite), 1010 (DeleteSiteView), 1011
(DeleteStore), 1012 (DeleteUserGroup), 1013
(DeleteVirtualHostingPool), 1014 (DeleteUpdateTask),
1015 (DeleteDiskUpdateDevice), 1016
(DeleteDiskVersion), 2001 (RunAddDeviceToDomain),
2002 (RunApplyAutoUpdate), 2003
(RunApplyIncrementalUpdate), 2004
(RunArchiveAuditTrail), 2005 (RunAssignAuthGroup),
2006 (RunAssignDevice), 2007 (RunAssignDiskLocator),
2008 (RunAssignServer), 2009 (RunWithReturnBoot),
2010 (RunCopyPasteDevice), 2011 (RunCopyPasteDisk),
2012 (RunCopyPasteServer), 2013 (RunCreateDirectory),
2014 (RunCreateDiskCancel), 2015
(RunDisableCollection), 2016 (RunDisableDevice), 2017
(RunDisableDeviceDiskLocator), 2018
(RunDisableDiskLocator), 2019 (RunDisableUserGroup),
2020 (RunDisableUserGroupDiskLocator), 2021
(RunWithReturnDisplayMessage), 2022
(RunEnableCollection), 2023 (RunEnableDevice), 2024
(RunEnableDeviceDiskLocator), 2025
(RunEnableDiskLocator), 2026 (RunEnableUserGroup),
2027 (RunEnableUserGroupDiskLocator), 2028
(RunExportOemLicenses), 2029 (RunImportDatabase),
2030 (RunImportDevices), 2031 (RunImportOemLicenses),
2032 (RunMarkDown), 2033 (RunWithReturnReboot), 2034
(RunRemoveAuthGroup), 2035 (RunRemoveDevice), 2036
(RunRemoveDeviceFromDomain), 2037
(RunRemoveDirectory), 2038 (RunRemoveDiskLocator),
2039 (RunResetDeviceForDomain), 2040
(RunResetDatabaseConnection), 2041
(RunRestartStreamingService), 2042
(RunWithReturnShutdown), 2043
(RunStartStreamingService), 2044
(RunStopStreamingService), 2045 (RunUnlockAllDisk),
2046 (RunUnlockDisk), 2047
(RunServerStoreVolumeAccess), 2048
(RunServerStoreVolumeMode), 2049 (RunMergeDisk), 2050
(RunRevertDiskVersion), 2051 (RunPromoteDiskVersion),
2052 (RunCancelDiskMaintenance), 2053
(RunActivateDevice), 2054 (RunAddDiskVersion), 2055
(RunExportDisk), 2056 (RunAssignDisk), 2057
(RunRemoveDisk), 2058 (RunDiskUpdateStart), 2059
(RunDiskUpdateCancel), 2060 (RunSetOverrideVersion),
2061 (RunCancelTask), 2062 (RunClearTask), 2063
(RunForceInventory), 2064 RunUpdateBDM, 2065
(RunStartDeviceDiskTempVersionMode), 2066
(RunStopDeviceDiskTempVersionMode), 3001
(RunWithReturnCreateDisk), 3002
(RunWithReturnCreateDiskStatus), 3003
(RunWithReturnMapDisk), 3004
(RunWithReturnRebalanceDevices), 3005
(RunWithReturnCreateMaintenanceVersion), 3006
(RunWithReturnImportDisk), 4001
(RunByteArrayInputImportDevices), 4002
(RunByteArrayInputImportOemLicenses), 5001
(RunByteArrayOutputArchiveAuditTrail), 5002
(RunByteArrayOutputExportOemLicenses), 6001

(SetAuthGroup), 6002 (SetCollection), 6003 (SetDevice), 6004 (SetDisk), 6005 (SetDiskLocator), 6006 (SetFarm), 6007 (SetFarmView), 6008 (SetServer), 6009 (SetServerBiosBootstrap), 6010 (SetServerBootstrap), 6011 (SetServerStore), 6012 (SetSite), 6013 (SetSiteView), 6014 (SetStore), 6015 (SetUserGroup), 6016 SetVirtualHostingPool, 6017 SetUpdateTask, 6018 SetDiskUpdateDevice, 7001 (SetListDeviceBootstraps), 7002 (SetListDeviceBootstrapsDelete), 7003 (SetListDeviceBootstrapsAdd), 7004 (SetListDeviceCustomProperty), 7005 (SetListDeviceCustomPropertyDelete), 7006 (SetListDeviceCustomPropertyAdd), 7007 (SetListDeviceDiskPrinters), 7008 (SetListDeviceDiskPrintersDelete), 7009 (SetListDeviceDiskPrintersAdd), 7010 (SetListDevicePersonality), 7011 (SetListDevicePersonalityDelete), 7012 (SetListDevicePersonalityAdd), 7013 (SetListDiskLocatorCustomProperty), 7014 (SetListDiskLocatorCustomPropertyDelete), 7015 (SetListDiskLocatorCustomPropertyAdd), 7016 (SetListServerCustomProperty), 7017 (SetListServerCustomPropertyDelete), 7018 (SetListServerCustomPropertyAdd), 7019 (SetListUserGroupCustomProperty), 7020 (SetListUserGroupCustomPropertyDelete), and 7021 (SetListUserGroupCustomPropertyAdd)

ObjectId: GUID of the object of the action. Default=00000000-0000-0000-0000-000000000000

ObjectName: Name of the object of the action. Default="" Max Length=1000

Path: Path of the object of the action. An example is Site\Collection for a Device. Default="" Max Length=101

SiteId: GUID of the Site for the object of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

SubId: GUID of the Collection or Store of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

ParentId: GUID of the parent action (one that triggered this action) if one exists. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

RootId: GUID of the root action (one that triggered this group of actions) if one exists. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

Attachments: An or'ed value that indicates if there are any details for this action. A value of 15 indicates that there are Children, Sibling, Parameters and Properties for

```
                    the action. Values are: 0 (None), 1 (Children), 2
                    (Sibling), 4 (Parameters), and 8 (Properties)
                    Default=0

    PvsAuditTrail[]: If successful, the PvsAuditTrail object(s) are
      returned.
```

EXAMPLE 1: Get PvsAuditTrail for Farm

```
    Get all PvsAuditTrail for the Farm.

    Get-PvsAuditTrail
```

EXAMPLE 2: Get PvsAuditTrail for FarmView

```
    Get all PvsAuditTrail for the FarmView named theFarmView.

    Get-PvsAuditTrail -FarmViewName theFarmView
```

EXAMPLE 3: Get PvsAuditTrail for Site

```
    Get all PvsAuditTrail for the Site named theSite.

    Get-PvsAuditTrail -SiteName theSite
```

EXAMPLE 4: Get PvsAuditTrail for SiteView

```
    Get all PvsAuditTrail for the SiteView named theSiteView in the Site
                    named theSite.

    Get-PvsAuditTrail -SiteViewName theSiteView -SiteName theSite

    SiteViewId can be used instead of SiteViewName so that the SiteName or
                    SiteId is not also needed.
```

EXAMPLE 5: Get PvsAuditTrail for Collection

```
    Get all PvsAuditTrail for the Collection named theCollection in the
                    Site named theSite.

    Get-PvsAuditTrail -CollectionName theCollection -SiteName theSite

    CollectionId can be used instead of CollectionName so that the
                    SiteName or SiteId is not also needed.
```

EXAMPLE 6: Get PvsAuditTrail for Device

```
    Get all PvsAuditTrail for the Device named theDevice.

    Get-PvsAuditTrail -DeviceName theDevice
```

EXAMPLE 7: Get PvsAuditTrail for Device MAC

```
    Get all PvsAuditTrail for the Device with MAC 02-50-F2-00-00-01.

    Get-PvsAuditTrail -DeviceMac "02-50-F2-00-00-01"
```

EXAMPLE 8: Get PvsAuditTrail for Server

```
    Get all PvsAuditTrail for the Server named theServer.

    Get-PvsAuditTrail -ServerName theServer
```

EXAMPLE 9: Get PvsAuditTrail for Store

```
    Get all PvsAuditTrail for the Store named theStore.

    Get-PvsAuditTrail -StoreName theStore
```

EXAMPLE 10: Get PvsAuditTrail for DiskLocator

```
Get all PvsAuditTrail for the DiskLocator named theDiskLocator in the
                Site named theSite and Store named theStore.

Get-PvsAuditTrail -DiskLocatorName theDiskLocator -SiteName theSite -
                StoreName theStore

DiskLocatorId can be used instead of DiskLocatorName so that the
                SiteName or SiteId and StoreName or StoreId are not
                also needed.
```

EXAMPLE 11: Get PvsAuditTrail for User

```
Get all PvsAuditTrail for the User named theUser in Domain theDomain.

Get-PvsAuditTrail -UserName theUser -DomainName theDomain
```

EXAMPLE 12: Get PvsAuditTrail for January

```
Get all PvsAuditTrail for January.

Get-PvsAuditTrail -BeginDate "01/01/2015 00:00" -EndDate "01/31/2015
                23:59"
```

EXAMPLE 13: Get PvsAuditTrail for DiskLocator and Server Type Actions

```
Get all PvsAuditTrail for 5 (DiskLocator) and 8 (Server) type actions.

Get-PvsAuditTrail -Type 5 8
```

EXAMPLE 14: Get PvsAuditTrail for DiskLocator and Server Type Actions

```
Get all PvsAuditTrail for the 1003 (DeleteDevice) and 1007
                (DeleteServer) actions.

Get-PvsAuditTrail -Action 1003 1007
```

EXAMPLE 15: Get PvsAuditTrail with Children

```
Get the 4fd16fc1-8dcc-4097-be5a-d0485bd7433b PvsAuditTrail and if it
                has children, the child PvsAuditTrail are retrieved.

$x = Get-PvsAuditTrail -AuditActionId "4fd16fc1-8dcc-4097-be5a-
                d0485bd7433b"

if (($x.Attachments -band 1) -eq 1) { Get-PvsAuditTrail -ParentId
                $x.AuditActionId }
```

Get-PvsAuthGroup

```
Get the fields for an AuthGroup, all AuthGroups in the system, AuthGroups
  with Farm, Site or Collection Authorization. All AuthGroups in the system
  are returned if no parameters are passed.

   One of these optional

   Guid[] Guid or AuthGroupId: GUID of the AuthGroup to Get.

   string[] Name or AuthGroupName: Name of the AuthGroup to Get.

   Guid[] SiteId: GUID of the Site to Get all AuthGroups with
                Authorization for.

   string[] SiteName: Name of the Site to Get all AuthGroups with
                Authorization for.
```

Guid[] CollectionId: GUID of the Collection to Get all AuthGroups with Authorization for.

or this optional & resolution

string[] CollectionName: Name of the Collection to Get all AuthGroups with Authorization for.

Optional

SwitchParameter Farm: If -Farm is specified, AuthGroups with Farm Authorization should be returned.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to Get all AuthGroups with Authorization for.

string[] SiteName: Name of the Site to Get all AuthGroups with Authorization for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

AuthGroupId, SiteId or CollectionId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or AuthGroupId: Read-only GUID that uniquely identifies this AuthGroup.

Name or AuthGroupName: Name of the Active Directory or Windows Group. Max Length=450

Description: User description. Default="" Max Length=250

Role: Role of the AuthGroup for a Collection. role can only be used with CollectionId or CollectionName. 300 is Collection Administrator, and 400 is Collection Operator. Default=999

PvsAuthGroup[]: If successful, the PvsAuthGroup object(s) are returned.

EXAMPLE 1: Get PvsAuthGroup for System

Get all PvsAuthGroup for the System.

```
Get-PvsAuthGroup
```

EXAMPLE 2: Get PvsAuthGroup for Farm

Get all PvsAuthGroup for the Farm.

```
Get-PvsAuthGroup
```

EXAMPLE 3: Get PvsAuthGroup for Site

Get all PvsAuthGroup for the Site named theSite.

```
Get-PvsAuthGroup -SiteName theSite
```

EXAMPLE 4: Get PvsAuthGroup for Collection

Get all PvsAuthGroup for the Collection named theCollection in the Site named theSite.

```
Get-PvsAuthGroup -CollectionName theCollection -SiteName theSite

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

Get-PvsAuthGroupUsage

```
Get the items that are authorized for an AuthGroup.

    One of these required

     Guid[] AuthGroupId: GUID of the AuthGroup to Get all items that are
                authorized for it.

     string[] Name or AuthGroupName: Name of the AuthGroup to Get all items
                that are authorized for it.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      AuthGroupId

    If only selected fields are needed, pass them in the Fields parameter
      as a string array.

     Guid or Id: GUID of the item. The item can be a Farm, Site or
                Collection. It will be 00000000-0000-0000-0000-
                000000000000 for Farm.

     Name: Name of the item. The item can be a Farm, Site or Collection.

     Role: Role of the AuthGroup for the item. 100 is Farm Administrator,
                200 is Site Administrator, 300 is Collection
                Administrator, and 400 is Collection Operator.
                Default=999

    PvsAuthGroupUsage[]: If successful, the PvsAuthGroupUsage object(s) are
      returned.
```

EXAMPLE 1: Get PvsAuthGroupUsage

```
     Get all PvsAuthGroupUsage for the AuthGroup named theAuthGroup.

     Get-PvsAuthGroupUsage -Name theAuthGroup
```

Get-PvsCeipData

```
Get the CEIP configuration

    Optional

     string[] Uuid: CEIP UUID of this Farm. This is optional since there is
                only one.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      Uuid

    If only selected fields are needed, pass them in the Fields parameter
      as a string array.

     Enabled: 1 if CEIP is enabled, otherwise 0. Min=0, Max=1
```

Uuid: CEIP UUID.

NextUpload: Date and time next CEIP upload is due if enabled is 1. Default=Empty

InProgress: 1 if an upload is currently in progress, otherwise 0. Default=0

ServerId: ID of server that is currently uploading, null if InProgress is 0. Default=00000000-0000-0000-0000-000000000000

OneTimeUpload: 1 to perform a one time upload. Default=0

PvsCeipData[]: If successful, the PvsCeipData object(s) are returned.

EXAMPLE 1: Get PvsCeipData for CeipData

Get all PvsCeipData for CeipData.

```
Get-PvsCeipData
```

Get-PvsCisData

Get the CIS configuration

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or CisDataId: CIS UUID

UserName: Username used to obtain the token Default="" Max Length=255

UploadToken: Token for uploading bundles to CIS Default="" Max Length=10

Path: Path where the last problem report bundle was saved Default="" Max Length=255

Password: Password of the user required to obtain the token. This is required only by Set and Add

PvsCisData[]: If successful, the PvsCisData object(s) are returned.

EXAMPLE 1: Get PvsCisData for CisData

Get all PvsCisData for CispData.

```
Get-PvsCisData
```

Get-PvsCollection

Get the fields for a Collection or all Collections in a Site or Farm. All Collections are returned if no parameters are passed.

One of these optional

Guid[] Guid or CollectionId: GUID of the Collection to Get.

Guid[] SiteId: GUID of the Site to Get all Collections for.

string[] SiteName: Name of the Site to Get all Collections for.

or this optional & resolution

string[] Name or CollectionName: Name of the Collection to Get.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to Get all Collections for.

string[] SiteName: Name of the Site to Get all Collections for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

CollectionId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or CollectionId: Read-only GUID that uniquely identifies this Collection.

Name or CollectionName: Name of the Collection. It is unique within the Site. Max Length=50

SiteId: GUID of the Site that this Collection is a member of. It is not used with SiteName.

SiteName: Name of the Site that this Collection is a member of. It is not used with SiteId.

Description: User description. Default="" Max Length=250

TemplateDeviceId: GUID of a Device in the Collection whose settings are used for initial values of new Devices. Not used with templateDeviceName. Default=00000000-0000-0000-0000-000000000000

TemplateDeviceName: Name of a Device in the Collection whose settings are used for initial values of new Devices. Not used with TemplateDeviceId. Default=""

LastAutoAddDeviceNumber: The Device Number of the last Auto Added Device. Default=0

Enabled: True when Devices in the Collection can be booted, false otherwise. Default=true

DeviceCount: Read-only count of Devices in this Collection. Default=0

DeviceWithPVDCount: Read-only count of Devices with Personal vDisk in this Collection. Default=0

ActiveDeviceCount: Read-only count of active Devices in this Collection. Default=0

MakActivateNeededCount: Read-only count of active Devices that need MAK activation in this Collection. Default=0

AutoAddPrefix: The string put before the Device Number for Auto Add. Default="" ASCII computer name characters no end digit Max Length=12

AutoAddSuffix: The string put after the Device Number for Auto Add. Default="" ASCII computer name characters no begin digit Max Length=12

AutoAddZeroFill: True when zeros be placed before the Device Number up to the AutoAddNumberLength for Auto Add, false otherwise. Default=true

AutoAddNumberLength: The maximum length of the Device Number for Auto Add. This length plus the AutoAddPrefix length plus the AutoAddSuffix length must be less than 16. Required that ((lenautoAddPrefix+lenautoAddSuffix)+AutoAddNumberLength)<=15. Min=3, Max=9, Default=4

Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 400 is Collection Operator. Default=999

PvsCollection[]: If successful, the PvsCollection object(s) are returned.

EXAMPLE 1: Get PvsCollection for Farm

Get all PvsCollection for the Farm.

```
Get-PvsCollection
```

EXAMPLE 2: Get PvsCollection for Site

Get all PvsCollection for the Site named theSite.

```
Get-PvsCollection -SiteName theSite
```

EXAMPLE 3: Get PvsCollection

Get the PvsCollection for the Collection named theCollection in the Site named theSite.

```
Get-PvsCollection -Name theCollection -SiteName theSite
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

EXAMPLE 4: Get PvsCollection and Enable

Get all PvsCollection that are not Enabled and then Enables them.

```
Get-PvsCollection -Fields Enabled | Where-Object {$_.Enabled -eq $false} | foreach { $o = $_; $o.Enabled = $true; $o } | Set-PvsCollection
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Get-PvsConnection

Return the PvsConnection object with the information about the SoapServer connection.

PvsConnection: If successful, the PvsConnection object is returned.

EXAMPLE 1: Get PvsConnection

Get the PvsConnection for the SoapServer.

```
Get-PvsConnection
```

Get-PvsCreateDiskStatus

Get the Percent Finished for an active CreateDisk. When finished, the PvsDiskLocator created is returned.

This required

string Name: Name of the Disk file that is being created.

One of these required

Guid StoreId: GUID of the Store that the Disk will be a member of.

string StoreName: Name of the Store that the Disk will be a member of.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

StoreId

UInt32 or PvsDiskLocator: If not finished, the percent complete is returned in an UInt32. If finished and successful, the PvsDiskLocator is returned.

EXAMPLE 1: Start-PvsCreateDiskStatus

This example shows how to use Get-PvsCreateDiskStatus during Start-PvsCreateDisk processing.

```
$thePvsDiskLocator = Start-PvsCreateDiskStatus -Name theDiskName -Size
                20480 -StoreName theStore -SiteName theSite -VHDX -
                Dynamic

while ($thePvsDiskLocator -eq $null)                                # while
                the create is processing

{

     %percentFinished = Get-PvsCreateDiskStatus -Name theDiskName -
                     StoreName theStore  # get percent finished or
                     DiskLocator when done

     if (%percentFinished.GetType().Name == "PvsDiskLocator")

     {

          $thePvsDiskLocator = %percentFinished

     }

     else

     {

          %percentFinished.ToString() + "% finished"                #
                          display percent finished

          Start-Sleep -seconds 10                                   #
                          wait 10 seconds more

     }

}

"Successful"
```

Get-PvsDevice

Get the fields for a Device, all Devices in a Collection, Site, Farm View, or Farm. All Devices are returned if no parameters are passed.

One of these optional

Guid[] Guid or DeviceId: GUID of the Device to Get.

string[] Name or DeviceName: Name of Device to Get.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Get.

Guid[] CollectionId: GUID of the Collection to Get all Devices for.

Guid[] ServerId: GUID of the Server to Get all Devices for.

string[] ServerName: Name of the Server to Get all Devices for.

Guid[] DiskLocatorId: GUID of the DiskLocator to Get all Devices for.

Guid[] SiteViewId: GUID of the Site View to Get all Devices for.

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Guid[] FarmViewId: GUID of the Farm View to Get all Devices for.

string[] FarmViewName: Name of the Farm View to Get all Devices for.

string[] BdmBoot: Include only the BDM Devices when set to 1. PXE devices if set to 0. If not included, all Devices are returned.

or one of these optional & resolutions

string[] CollectionName: Name of the Collection to Get all Devices for.

string[] DiskLocatorName: Name of the DiskLocator to Get all Devices for.

string[] SiteViewName: Name of the Site View to Get all Devices for.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceName, CollectionId, ServerId, DiskLocatorId, SiteViewId, SiteId or FarmViewId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or DeviceId: Read-only GUID that uniquely identifies this Device.

Name or DeviceName: Computer name with no spaces. ASCII computer name characters Max Length=15

CollectionId: GUID of the Collection this Device is to be a member of. It is not used with CollectionName.

CollectionName: Name of the Collection this Device is to be a member of. SiteName or SiteId must also be used.

SiteId: GUID of the Site the CollectionName is to be a member of. This or SiteName is used with CollectionName.

SiteName: Name of the Site the CollectionName is to be a member of. This or SiteId is used with CollectionName.

Description: User description. Default="" Max Length=250

DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device.

BootFrom: Device to boot from. Choices are 1 for vDisk, 2 for Hard Disk, and 3 for Floppy. This cannot be Set for a Device with Personal vDisk. Min=1, Max=3, Default=1

ClassName: Used by Automatic Update feature to match new versions of Disks to a Device. This cannot be Set for a Device with Personal vDisk. Default="" Max Length=41

Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

Enabled: True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true

LocalDiskEnabled: If there is a local disk menu choice for the Device, this is true. This cannot be Set for a Device with Personal vDisk. Default=false

Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 400 is Collection Operator. Default=999

Authentication: Device log in authentication. Choices are 0 for none, 1 for User Name/Password, and 2 for Extern. This cannot be Set for a Device with Personal vDisk. Min=0, Max=2, Default=0

User: Name of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=20

Password: Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=100

Active: True if the Device is currently active, false otherwise. Default=false

Template: True if the Device is the template in its Collection, false otherwise. Default=false

AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" ASCII Max Length=256

LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

Type: 1 when it performs test of Disks, 2 when it performs maintenance on Disks, 3 when it has a Personal vDisk, 4 when it has a Personal vDisk and performs tests, 0 otherwise. Min=0, Max=4, Default=0

PvdDriveLetter: Read-only Personal vDisk Drive letter. Range is E to U and W to Z. Default="" Max Length=1

LocalWriteCacheDiskSize: The size in GB to format the Device cache file disk. If the value is 0, then the disk is not formatted. Min=0, Max=2048, Default=0

VirtualHostingPoolId: GUID that uniquely identifies the Virtual Hosting Pool for a VM. This is needed when Adding a VM device. Default=00000000-0000-0000-0000-000000000000

HypVmId: Hypervisor VM ID for HCL Default="" Max Length=250

TemporaryVersionSet: Read-only true when temporary version is set. Default=false

BdmBoot: Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false

BdmType: Use PXE boot when set to 0, BDM (Bios) boot when set to 1 and BDM (Uefi) boot when set to 2. Default=0

BdmFormat: 1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0

BdmUpdated: Timestamp of the last BDM boot disk update. Default=Empty

BdmCreated: Timstamp when BDM device was created Default=Empty

XsPvsProxyUuid: UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000

PvsDevice[]: If successful, the PvsDevice object(s) are returned.

EXAMPLE 1: Get PvsDevice for Farm

Get all PvsDevice for the Farm.

```
Get-PvsDevice
```

EXAMPLE 2: Get PvsDevice for FarmView

Get all PvsDevice for the FarmView named theFarmView.

```
Get-PvsDevice -FarmViewName theFarmView
```

EXAMPLE 3: Get PvsDevice for Site

Get all PvsDevice for the Site named theSite.

```
Get-PvsDevice -SiteName theSite
```

EXAMPLE 4: Get PvsDevice for SiteView

Get all PvsDevice for the SiteView named theSiteView in the Site named theSite.

```
Get-PvsDevice -SiteViewName theSiteView -SiteName theSite
```

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

EXAMPLE 5: Get PvsDevice for Collection

Get all PvsDevice for the Collection named theCollection in the Site named theSite.

```
Get-PvsDevice -CollectionName theCollection -SiteName theSite
```

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 6: Get PvsDevice for DiskLocator

Get all PvsDevice for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
Get-PvsDevice -DiskLocatorName theDiskLocator -SiteName theSite -StoreName theStore
```

DiskLocatorId can be used instead of DiskLocatorName so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 7: Get PvsDevice for Device

Get the PvsDevice for the Device named theDevice.

```
Get-PvsDevice -Name theDevice
```

EXAMPLE 8: Get PvsDevice and Enable

Get all PvsDevice that are not Enabled and then Enables them.

```
Get-PvsDevice -Fields Enabled | Where-Object {$_.Enabled -eq $false} |
                foreach { $o = $_; $o.Enabled = $true; $o } | Set-
                PvsDevice
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Get-PvsDeviceBootstrap

Get all Bootstrap files for a Device, and the MenuText for each.

One of these required

Guid[] Guid or DeviceId: GUID of the Device.

string[] Name or DeviceName: Name of the Device.

PvsPhysicalAddress[] DeviceMac: MAC of the Device.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId

If only selected fields are needed, pass them in the Fields parameter as a string array.

These fields exist in the DeviceBootstrap array within each PvsDeviceBootstrap returned.

Each array item is a PvsDeviceBootstrapList object.

Name or Bootstrap: Name of the bootstrap file. Max Length=259

MenuText: Text that is displayed in the Boot Menu. If this field has no value, the bootstrap value is used. Default="" ASCII Max Length=64

PvsDeviceBootstrap[]: If successful, the PvsDeviceBootstrap object(s) are returned.

EXAMPLE 1: Get PvsDeviceBootstrap for Device

Get all PvsDeviceBootstrap for the Device named theDevice.

```
Get-PvsDeviceBootstrap -Name theDevice
```

EXAMPLE 2: Get PvsDeviceBootstrap for Device MAC

Get all PvsDeviceBootstrap for the Device with MAC 02-50-F2-00-00-01.

```
Get-PvsDeviceBootstrap -DeviceMac "02-50-F2-00-00-01"
```

Get-PvsDeviceCount

Get count of Devices in a Collection or View.

One of these required

Guid CollectionId: GUID of the Collection to get the Device Count of.

Guid SiteViewId: GUID of the Site View to get the Device Count of.

Guid FarmViewId: GUID of the Farm View to get the Device Count of.

string FarmViewName: Name of the Farm View to get the Device Count of.

or one of these required & resolutions

string CollectionName: Name of the Collection to get the Device Count of.

string SiteViewName: Name of the Site View to get the Device Count of.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

CollectionId, SiteViewId or FarmViewId

UInt32: If successful, the numeric value is returned

EXAMPLE 1: Get-PvsDeviceCount Returns the Number (or Count) of PvsDevice in PvsCollection

Get-PvsDeviceCount -CollectionName theCollection -SiteName theSite

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Get-PvsDeviceCount Returns the Number (or Count) of PvsDevice in PvsFarmView

Get-PvsDeviceCount -FarmViewName theFarmView

EXAMPLE 3: Get-PvsDeviceCount Returns the Number (or Count) of PvsDevice in PvsSiteView

Get-PvsDeviceCount -SiteViewName theSiteView -SiteName theSite

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

Get-PvsDeviceDiskLocatorEnabled

Return true if a Device/DiskLocator is enabled.

One of these required

Guid Guid or DeviceId: Device GUID, to see if the DiskLocator for it is enabled.

string Name or DeviceName: Device name, to see if the DiskLocator for it is enabled.

PvsPhysicalAddress[] DeviceMac: MAC of the Device, to see if the DiskLocator for it is enabled.

This required

Guid DiskLocatorId: DiskLocator GUID, to see if it is enabled for the Device.

or this required & resolution

string DiskLocatorName: DiskLocator name, to see if it is enabled for the Device.

One of these resolutions when needed

Guid StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId or DiskLocatorId

String: If successful, the String value is returned.

EXAMPLE 1: Get-PvsDeviceDiskLocatorEnabled Determine if PvsDevice/PvsDiskLocator is Enabled

```
Get-PvsDeviceDiskLocatorEnabled -Name theDevice -DiskLocatorName
theDiskLocator -StoreName theStore
```

DiskLocatorId can be used instead of DiskLocatorName so that the StoreName or StoreId are not also needed.

Get-PvsDeviceDiskTempVersion

Get Temporary Disk Version information for a Device, DiskLocator, Disk Version, Site or Farm.

One of these optional

Guid[] Guid or DeviceId: GUID of the Device to get the temporary disk version information for.

string[] Name or DeviceName: Name of the Device to get the temporary disk version information for.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to get the temporary disk version information for.

Guid[] SiteId: GUID of the Site to get temporary disk version information for, and also resolution for DiskLocatorName.

string[] SiteName: Name of the Site to get temporary disk version information for, and also resolution for DiskLocatorName.

Guid[] DiskLocatorId: GUID of the Disk Locator to get temporary disk version information for.

or this optional & resolution

string[] DiskLocatorName: Name of the Disk Locator to get temporary disk version information for.

This optional & resolution

string[] Version: Version of the DiskLocator specified to get temporary disk version information for. Needs the DiskLocatorId or DiskLocatorName too.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to get temporary disk version information for, and also resolution for DiskLocatorName.

string[] SiteName: Name of the Site to get temporary disk version information for, and also resolution for DiskLocatorName.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId, SiteId or DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or DeviceId: Read-only GUID that uniquely identifies the Device with temporary version.

Name or DeviceName: Read-only Computer name that uniquely identifies the Device with temporary version. ASCII computer name characters

DiskLocatorId: Read-only GUID that uniquely identifies then Disk Locator with temporary version.

DiskLocatorName: Read-only Name of the Disk Locator File with temporary version. It is unique within the Store. ASCII

SiteId: Read-only GUID of the Site the Device and DiskLocator are a member of.

SiteName: Read-only Name of the Site the Device and DiskLocator are a member of.

StoreId: Read-only GUID of the Store that the Disk Locator is a member of.

StoreName: Read-only Name of the Store that the Disk Locator is a member of.

Version: Read-only Disk version the temporary is for.

PvsDeviceDiskTempVersion[]: If successful, the PvsDeviceDiskTempVersion object(s) are returned.

EXAMPLE 1: Get PvsDeviceDiskTempVersion for Device

Get the PvsDeviceDiskTempVersion for the Device named theDevice.

```
Get-PvsDeviceDiskTempVersion -Name theDevice
```

EXAMPLE 2: Get PvsDeviceDiskTempVersion for DiskLocator

```
Get all PvsDeviceDiskTempVersion for the DiskLocator named
                theDiskLocator in the Site named theSite and Store
                named theStore.

Get-PvsDeviceDiskTempVersion -DiskLocatorName theDiskLocator -SiteName
                theSite -StoreName theStore

DiskLocatorId can be used instead of DiskLocatorName so that the
                SiteName or SiteId and StoreName or StoreId are not
                also needed.
```

EXAMPLE 3: Get PvsDeviceDiskTempVersion for DiskLocator with Version

```
Get-PvsDeviceDiskTempVersion -DiskLocatorName theDiskLocator -Version
                4 -SiteName theSite -StoreName theStore

DiskLocatorId can be used instead of DiskLocatorName so that the
                StoreName or StoreId are not also needed.
```

EXAMPLE 4: Get PvsDeviceDiskTempVersion for Site

```
Get all PvsDeviceDiskTempVersion for the Site named theSite.

Get-PvsDeviceDiskTempVersion -SiteName theSite
```

EXAMPLE 5: Get PvsDeviceDiskTempVersion for Farm

```
Get all PvsDeviceDiskTempVersion for the Farm.

Get-PvsDeviceDiskTempVersion
```

Get-PvsDeviceInfo

```
Get the fields and status for a Device, all Devices in a Collection, Site,
   Farm View, or Farm. All Devices are returned if no parameters are passed.

    One of these optional

     Guid[] Guid or DeviceId: GUID of the Device to Get.

     string[] Name or DeviceName: Name of Device to Get.

     PvsPhysicalAddress[] DeviceMac: MAC of the Device to Get.

     Guid[] CollectionId: GUID of the Collection to Get all Devices for.

     Guid[] ServerId: GUID of the Server to Get all Devices for.

     string[] ServerName: Name of the Server to Get all Devices for.

     Guid[] DiskLocatorId: GUID of the DiskLocator to Get all Devices for.

     Guid[] SiteViewId: GUID of the Site View to Get all Devices for.

     Guid[] SiteId: GUID of the Site.

     string[] SiteName: Name of the Site.

     Guid[] FarmViewId: GUID of the Farm View to Get all Devices for.

     string[] FarmViewName: Name of the Farm View to Get all Devices for.

     string[] BdmBoot: Include only the BDM Devices when set to 1. PXE
                devices if set to 0. If not included, all Devices are
                returned.
```

or one of these optional & resolutions

string[] CollectionName: Name of the Collection to Get all Devices for.

string[] DiskLocatorName: Name of the DiskLocator to Get all Devices for.

string[] SiteViewName: Name of the Site View to Get all Devices for.

Optional

SwitchParameter OnlyActive: If -OnlyActive is specified, include only the active Devices, otherwise all Devices are returned. Only active Devices are always returned for ServerId, ServerName, or version.

uint MakLicenseActivated: Optional MAK licensing indicator value to only return active Devices for. Values are: 0 (MAK not used), 1 (Not Activated), 2 (Activated).

uint Version: Version of the Disk to Get all active Devices for. This is used with DiskLocatorId or DiskLocatorName.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceName, CollectionId, ServerId, DiskLocatorId, SiteViewId, SiteId or FarmViewId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or DeviceId: Read-only GUID that uniquely identifies this Device.

Name or DeviceName: Computer name with no spaces. ASCII computer name characters Max Length=15

CollectionId: GUID of the Collection this Device is to be a member of. It is not used with CollectionName.

CollectionName: Name of the Collection this Device is to be a member of. SiteName or SiteId must also be used.

SiteId: GUID of the Site the CollectionName is to be a member of. This or SiteName is used with CollectionName.

SiteName: Name of the Site the CollectionName is to be a member of. This or SiteId is used with CollectionName.

Description: User description. Default="" Max Length=250

DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device.

BootFrom: Device to boot from. Choices are 1 for vDisk, 2 for Hard Disk, and 3 for Floppy. This cannot be Set for a Device with Personal vDisk. Min=1, Max=3, Default=1

ClassName: Used by Automatic Update feature to match new versions of Disks to a Device. This cannot be Set for a Device with Personal vDisk. Default="" Max Length=41

Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

Enabled: True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true

LocalDiskEnabled: If there is a local disk menu choice for the Device, this is true. This cannot be Set for a Device with Personal vDisk. Default=false

Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 400 is Collection Operator. Default=999

Authentication: Device log in authentication. Choices are 0 for none, 1 for User Name/Password, and 2 for Extern. This cannot be Set for a Device with Personal vDisk. Min=0, Max=2, Default=0

User: Name of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=20

Password: Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=100

Active: True if the Device is currently active, false otherwise. Default=false

Template: True if the Device is the template in its Collection, false otherwise. Default=false

AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" ASCII Max Length=256

LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

Type: 1 when it performs test of Disks, 2 when it performs maintenance on Disks, 3 when it has a Personal vDisk, 4 when it has a Personal vDisk and performs tests, 0 otherwise. Min=0, Max=4, Default=0

PvdDriveLetter: Read-only Personal vDisk Drive letter. Range is E to U and W to Z. Default="" Max Length=1

LocalWriteCacheDiskSize: The size in GB to format the Device cache file disk. If the value is 0, then the disk is not formatted. Min=0, Max=2048, Default=0

VirtualHostingPoolId: GUID that uniquely identifies the Virtual Hosting Pool for a VM. This is needed when Adding a VM device. Default=00000000-0000-0000-0000-000000000000

HypVmId: Hypervisor VM ID for HCL Default="" Max Length=250

TemporaryVersionSet: Read-only true when temporary version is set. Default=false

BdmBoot: Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false

BdmType: Use PXE boot when set to 0, BDM (Bios) boot when set to 1 and BDM (Uefi) boot when set to 2. Default=0

BdmFormat: 1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0

BdmUpdated: Timestamp of the last BDM boot disk update. Default=Empty

BdmCreated: Timstamp when BDM device was created Default=Empty

XsPvsProxyUuid: UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000

Ip: Read-only IP of the Device. It is equal to 0.0.0.0 if the Device is not active.

ServerPortConnection: Read-only Port of the Server that the Device is using. It is equal to 0 if the Device is not active. Default=0

ServerIpConnection: Read-only IP of the Server that the Device is using. It is equal to 0.0.0.0 if the Device is not active.

ServerId: Read-only GUID of the Server that the Device is using. It is equal to 00000000-0000-0000-0000-000000000000 if the Device is not active.

ServerName: Read-only Name of the Server that the Device is using. It is equal to "" if the Device is not active.

DiskLocatorId: Read-only GUID of the Disk Locator that the Device is using. It is equal to 00000000-0000-0000-0000-000000000000 if the Device is not active.

DiskLocatorName: Read-only name of the Disk Locator File that the Device is using. It is equal to the list of Disk Locator names for the Device if the Device is not active.

DiskVersion: Read-only version of the Disk Locator File that the Device is using. It is equal to 0 if the Device is not active. Default=0

DiskVersionAccess: State of the Disk Version. Values are: 0 (Production), 1 (Maintenance), 2 (MaintenanceHighestVersion), 3 (Override), 4 (Merge), 5 (MergeMaintenance), 6 (MergeTest), and 7 (Test). It is equal to 0 if the Device is not active. Default=0

DiskFileName: Name of the Disk File including the extension. It is equal to "" if the Device is not active.

Status: 1 or 2 numbers in the format n,n. They are the number of retries and if ram cache is being used, ram cache percent used. It is equal to "" if the Device is not active.

LicenseType: 0 when None, 1 for Desktop, 2 for Server, 5 for OEM SmartClient, 6 for XenApp, 7 for XenDesktop. It is equal to 0 if the Device is not active. Default=0

MakLicenseActivated: Read-only indicator if MAK licensing is being used and is activated. Values are: 0 (MAK not used), 1 (Not Activated), 2 (Activated). It is equal to 0 if the Device is not active. Default=0

Model: Oem Only: Read-only model of the computer. Values are OptiPlex 745, 755, 320, 760, FX160, or Default. It is equal to "" if the Device is not active.

License: Oem Only: Read-only type of the license. Values are 0 when None, 1 or 2 when Desktop. It is equal to 0 if the Device is not active. Default=0

PvsDeviceInfo[]: If successful, the PvsDeviceInfo object(s) are returned.

EXAMPLE 1: Get PvsDeviceInfo for Farm

Get all PvsDeviceInfo for the Farm.

```
Get-PvsDeviceInfo
```

EXAMPLE 2: Get PvsDeviceInfo for FarmView

Get all PvsDeviceInfo for the FarmView named theFarmView.

```
Get-PvsDeviceInfo -FarmViewName theFarmView
```

EXAMPLE 3: Get PvsDeviceInfo for Site

Get all PvsDeviceInfo for the Site named theSite.

```
Get-PvsDeviceInfo -SiteName theSite
```

EXAMPLE 4: Get PvsDeviceInfo for SiteView

```
Get all PvsDeviceInfo for the SiteView named theSiteView in the Site
                named theSite.

Get-PvsDeviceInfo -SiteViewName theSiteView -SiteName theSite

SiteViewId can be used instead of SiteViewName so that the SiteName or
                SiteId is not also needed.
```

EXAMPLE 5: Get PvsDeviceInfo for Collection

```
Get all PvsDeviceInfo for the Collection named theCollection in the
                Site named theSite.

Get-PvsDeviceInfo -CollectionName theCollection -SiteName theSite

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 6: Get PvsDeviceInfo for DiskLocator

```
Get all PvsDeviceInfo for the DiskLocator named theDiskLocator in the
                Site named theSite and Store named theStore.

Get-PvsDeviceInfo -DiskLocatorName theDiskLocator -SiteName theSite -
                StoreName theStore

DiskLocatorId can be used instead of DiskLocatorName so that the
                SiteName or SiteId and StoreName or StoreId are not
                also needed.
```

EXAMPLE 7: Get PvsDeviceInfo for Device

```
Get the PvsDeviceInfo for the Device named theDevice.

Get-PvsDeviceInfo -Name theDevice
```

EXAMPLE 8: Get PvsDeviceInfo and Enable

```
Get all PvsDeviceInfo that are not Enabled and then Enables them.

Get-PvsDeviceInfo -Fields Enabled | Where-Object {$_.Enabled -eq
                $false} | foreach { $o = $_; $o.Enabled = $true; $o }
                | Set-PvsDevice

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y
                and returns the object again so it can be piped to
                the Set command for update.
```

Get-PvsDevicePersonality

```
Get the Device Personality names and values.

    One of these required

    Guid[] Guid or DeviceId: GUID of the Device.

    string[] Name or DeviceName: Name of the Device.

    PvsPhysicalAddress[] DeviceMac: MAC of the Device.
```

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId

If only selected fields are needed, pass them in the Fields parameter as a string array.

These fields exist in the DevicePersonality array within each PvsDevicePersonality returned.

Each array item is a PvsDevicePersonalityList object.

Name: Name of the Device personality item. Max Length=250

Value: Value for the Device personality item. Max Length=1000

PvsDevicePersonality[]: If successful, the PvsDevicePersonality object(s) are returned.

EXAMPLE 1: Get PvsDevicePersonality for Device

Get all PvsDevicePersonality for the Device named theDevice.

```
Get-PvsDevicePersonality -Name theDevice
```

EXAMPLE 2: Get PvsDevicePersonality for Device MAC

Get all PvsDevicePersonality for the Device with MAC 02-50-F2-00-00-01.

```
Get-PvsDevicePersonality -DeviceMac "02-50-F2-00-00-01"
```

Get-PvsDeviceStatus

Get the DeviceStatus fields for a Device or all Devices for a Server, Disk Locator, or Farm. All Devices are returned if no parameters are passed.

One of these optional

Guid[] Guid or DeviceId: GUID of the Device to Get status for.

string[] Name or DeviceName: Name of Device to Get status for.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Get status for.

Guid[] ServerId: GUID of the Server to Get all Device Status for.

string[] ServerName: Name of the Server to Get all Device Status for.

Guid[] DiskLocatorId: GUID of the Disk Locator to Get all DeviceStatus for.

Guid[] CollectionId: GUID of the Collection to Get all DeviceStatus for.

or one of these optional & resolutions

string[] DiskLocatorName: Name of the Disk Locator File to Get all DeviceStatus for.

string[] CollectionName: Name of the Collection to Get all DeviceStatus for.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId, ServerId, DiskLocatorId or CollectionId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or DeviceId: Read-only GUID of the Device. Can be used with Get Device.

Name or DeviceName: Read-only Name of the Device. Can be used with Get Device.

Ip: Read-only IP of the Device.

ServerPortConnection: Read-only Port of the Server that the Device is using. Default=0

ServerIpConnection: Read-only IP of the Server that the Device is using.

ServerId: Read-only GUID of the Server that the Device is using.

ServerName: Read-only Name of the Server that the Device is using.

DiskLocatorId: Read-only GUID of the Disk Locator that the Device is using.

DiskLocatorName: Read-only name of the Disk Locator File that the Device is using.

DiskVersion: Read-only version of the Disk Locator File that the Device is using. Default=0

DiskVersionAccess: State of the Disk Version. Values are: 0 (Production), 1 (Maintenance), 2 (MaintenanceHighestVersion), 3 (Override), 4 (Merge), 5 (MergeMaintenance), 6 (MergeTest), and 7 (Test) Default=0

DiskFileName: Name of the Disk File including the extension.

Status: 1 or 2 numbers in the format n,n. They are the number of retries and if ram cache is being used, ram cache percent used.

LicenseType: 0 when None, 1 for Desktop, 2 for Server, 5 for OEM SmartClient, 6 for XenApp, 7 for XenDesktop. Default=0

MakLicenseActivated: Read-only indicator if MAK licensing is being used and is activated. Values are: 0 (MAK not used), 1 (Not Activated), 2 (Activated). Default=0

PvsDeviceStatus[]: If successful, the PvsDeviceStatus object(s) are returned.

EXAMPLE 1: Get PvsDeviceStatus for Farm

Get all PvsDeviceStatus for the Farm.

```
Get-PvsDeviceStatus
```

EXAMPLE 2: Get PvsDeviceStatus for FarmView

Get all PvsDeviceStatus for the FarmView named theFarmView.

```
Get-PvsDeviceStatus -FarmViewName theFarmView
```

EXAMPLE 3: Get PvsDeviceStatus for Site

Get all PvsDeviceStatus for the Site named theSite.

```
Get-PvsDeviceStatus -SiteName theSite
```

EXAMPLE 4: Get PvsDeviceStatus for SiteView

Get all PvsDeviceStatus for the SiteView named theSiteView in the Site named theSite.

```
Get-PvsDeviceStatus -SiteViewName theSiteView -SiteName theSite
```

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

EXAMPLE 5: Get PvsDeviceStatus for Collection

Get all PvsDeviceStatus for the Collection named theCollection in the Site named theSite.

```
Get-PvsDeviceStatus -CollectionName theCollection -SiteName theSite
```

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 6: Get PvsDeviceStatus for DiskLocator

Get all PvsDeviceStatus for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
Get-PvsDeviceStatus -DiskLocatorName theDiskLocator -SiteName theSite
-StoreName theStore
```

DiskLocatorId can be used instead of DiskLocatorName so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 7: Get PvsDeviceStatus for Device

Get the PvsDeviceStatus for the Device named theDevice.

```
Get-PvsDeviceStatus -Name theDevice
```

Get-PvsDirectory

Look for Directories or Drives on the Server specified. Return a String array of the Directories or Drives found.

One of these required

Guid Guid or ServerId: GUID of the Server to get a list of Directories or Drives.

string Name or ServerName: Name of the Server to get a list of Directories or Drives.

Optional

string Path: Path to get list of Directories for. If not specified, the Drives are returned.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

string[]: If successful, the array of directory names is returned.

EXAMPLE 1: Get-PvsDirectory for Name

```
Get-PvsDirectory -Name theServer -Path "C:\directory"
```

EXAMPLE 2: Get-PvsDirectory for PvsServer Using Pipe

The Get-PvsServer output is piped to the Get-PvsDirectory.

```
Get-PvsServer -Name theServer -Fields Guid | Get-PvsDirectory -Path "C:\directory"
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Get-PvsDisk

Get the fields for a single disk.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Class: Class of the Disk. Max Length=40

ImageType: Type of this image (software type). Max Length=40

DiskSize: Read-only size of the image. The value is 0 when it is not available. Default=0

VhdBlockSize: Block size in KB. For VHD it is only used with Dynamic type. Tested sizes for VHD are 512, 2048, and 16384. VHD Min=512, Max=16384, Default=2048. For VHDX it is used for all types. Tested size for VHDX is 32768. VHDX Min=1024, Max= 262144, Default=32768. Default=0

LogicalSectorSize: Logical Sector Size. Values are: 512, 4096, Default=512

WriteCacheSize: RAM cache size (MB). Not 0 when used with Cache in Device RAM, and Cache in Device RAM with Overflow on Hard Disk. A value of 0 will disable the RAM use for Cache in Device RAM with Overflow on Hard Disk. Min=0, Max=131072, Default=0

AutoUpdateEnabled: Automatically update this image for matching Devices when set to true. Default false

ActivationDateEnabled: Use activation date to activate image when set to true. Default false

AdPasswordEnabled: Enable AD password management when set to true.

HaEnabled: Enable HA when set to true.

PrinterManagementEnabled: Invalid printers will be deleted from the Device when set to true.

WriteCacheType: 0 (Private), (other values are standard image) 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), or 9 (Cache in Device RAM with Overflow on Hard Disk). Min=0, Max=9, Default=0

LicenseMode: 0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0

ActiveDate: Date to activate the disk if AutoUpdateEnabled and activationDateEnabled are true. Has the date. Empty when the AutoUpdateEnabled or activationDateEnabled are false.

LongDescription: Description of the Disk. Max Length=399

OperatingSystem: Operating System of Disk. Max Length=250

OsType: Operating System Type of Disk. Max Length=40

SerialNumber: User defined serial number. Max Length=36

Date: User defined date. Max Length=40

Author: User defined author. Max Length=40

Title: User defined title. Max Length=40

Company: User defined company. Max Length=40

InternalName: User defined name. Max Length=63

OriginalFile: User defined original file. Max Length=127

HardwareTarget: User defined hardware target. Max Length=127

MajorRelease: User defined major release number. Min=0, Max=4294967295, Default=0

MinorRelease: User defined minor release number. Min=0, Max=4294967295, Default=0

Build: User defined build number. Min=0, Max=4294967295, Default=0

ClearCacheDisabled: Clear cached secrets disabled.

VHDX: If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false

PvsDisk[]: If successful, the PvsDisk object(s) are returned.

EXAMPLE 1: Get PvsDisk for DiskLocator

Get the PvsDisk for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
Get-PvsDisk -Name theDiskLocator -SiteName theSite -StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Get-PvsDiskInfo

Get the fields for a Disk and Disk Locator or all Disks and Disk Locators for a Device, Server, Store, Site, or Farm. All Disks and DiskLocators are returned if no parameters are passed.

One of these optional

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Get.

Guid[] DeviceId: GUID of the Device to Get all DiskLocators for.

string[] DeviceName: Name of the Device to Get all DiskLocators for.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Get all DiskLocators for.

Guid[] ServerId: GUID of the Server to Get all DiskLocators for.

string[] ServerName: Name of the Server to Get all DiskLocators for.

Guid[] UpdateTaskId: GUID of the Update Task to Get all DiskLocators for.

Guid[] SiteId: GUID of the Site to Get all DiskLocators for.

string[] SiteName: Name of the Site to Get all DiskLocators for.

or one of these optional & resolutions

string[] Name or DiskLocatorName: Name of the Disk Locator File to Get.

Guid[] StoreId: GUID of the Store to Get all DiskLocators for.

string[] StoreName: Name of the Store to Get all DiskLocators for.

string[] UpdateTaskName: Name of the Update Task to Get all DiskLocators for.

One of these optional

SwitchParameter Single: If -Single is specified, include single server connection. If this and All are not included, both connection types are included.

SwitchParameter All: If -All is specified, include all server connections for the store. If this and Single are not included, both connection types are included.

Optional

SwitchParameter OnlyActive: If -OnlyActive is specified, include only the active DiskLocators. If not included, all DiskLocators are returned.

SwitchParameter UpdateDevice: If -UpdateDevice is specified, include only DiskLocators that have an Update Device.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to Get all DiskLocators for.

string[] SiteName: Name of the Site to Get all DiskLocators for.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store to Get all DiskLocators for.

string[] StoreName: Name of the Store to Get all DiskLocators for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId, DeviceId, ServerId, UpdateTaskId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or DiskLocatorId: Read-only GUID that uniquely identifies this Disk Locator.

Name or DiskLocatorName: Name of the Disk Locator File. It is unique within the Store. ASCII Max Length=52

SiteId: GUID of the Site this DiskLocator is to be a member of. It is not used with SiteName.

SiteName: Name of the Site this DiskLocator is to be a member of. It is not used with SiteId.

StoreId: GUID of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreName.

StoreName: Name of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreId.

Description: User description. Default="" Max Length=250

MenuText: Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default="" ASCII Max Length=64

ServerId: GUID of the single Server that this Disk Locator is assigned to. It is not used with ServerName. Default=00000000-0000-0000-0000-000000000000

ServerName: Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=""

Enabled: True when this disk can be booted, false otherwise. Default=true

Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 999 is read-only. Default=999

Mapped: True if the Disk is currently mapped, false otherwise. Default=false

EnabledForDevice: True when this disk is enabled for the Device specified, false otherwise. This is only returned when a Device is specified. Default=true

Active: True if the DiskLocator is currently active, false otherwise. Default=false

RebalanceEnabled: True when this Server can automatically rebalance Devices, false otherwise. Default=false

RebalanceTriggerPercent: Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25

SubnetAffinity: Qualifier for subnet affinity when assigning a Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2, Default=0

DiskUpdateDeviceId: GUID of the DiskUpdateDevice that is used when updates are performed. Default=00000000-0000-0000-0000-000000000000

DiskUpdateDeviceName: Name of the DiskUpdateDevice that is used when updates are performed. Default=""

TemporaryVersionSet: Read-only true when temporary version(s) are set. Default=false

Class: Class of the Disk. Max Length=40

ImageType: Type of this image (software type). Max Length=40

DiskSize: Read-only size of the image. The value is 0 when it is not available. Default=0

VhdBlockSize: Block size in KB. For VHD it is only used with Dynamic type. Tested sizes for VHD are 512, 2048, and 16384. VHD Min=512, Max=16384, Default=2048. For VHDX it is used for all types. Tested size for VHDX is 32768. VHDX Min=1024, Max= 262144, Default=32768. Default=0

LogicalSectorSize: Logical Sector Size. Values are: 512, 4096, Default=512

WriteCacheSize: RAM cache size (MB). Not 0 when used with Cache in Device RAM, and Cache in Device RAM with Overflow on Hard Disk. A value of 0 will disable the RAM use for Cache in Device RAM with Overflow on Hard Disk. Min=0, Max=131072, Default=0

AutoUpdateEnabled: Automatically update this image for matching Devices when set to true. Default false

ActivationDateEnabled: Use activation date to activate image when set to true. Default false

AdPasswordEnabled: Enable AD password management when set to true.

HaEnabled: Enable HA when set to true.

PrinterManagementEnabled: Invalid printers will be deleted from the Device when set to true.

WriteCacheType: 0 (Private), (other values are standard image) 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), or 9 (Cache in Device RAM with Overflow on Hard Disk). Min=0, Max=9, Default=0

LicenseMode: 0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0

ActiveDate: Date to activate the disk if AutoUpdateEnabled and activationDateEnabled are true. Has the date. Empty when the AutoUpdateEnabled or activationDateEnabled are false.

LongDescription: Description of the Disk. Max Length=399

OperatingSystem: Operating System of Disk. Max Length=250

OsType: Operating System Type of Disk. Max Length=40

SerialNumber: User defined serial number. Max Length=36

Date: User defined date. Max Length=40

Author: User defined author. Max Length=40

Title: User defined title. Max Length=40

Company: User defined company. Max Length=40

InternalName: User defined name. Max Length=63

OriginalFile: User defined original file. Max Length=127

HardwareTarget: User defined hardware target. Max Length=127

MajorRelease: User defined major release number. Min=0, Max=4294967295, Default=0

MinorRelease: User defined minor release number. Min=0, Max=4294967295, Default=0

Build: User defined build number. Min=0, Max=4294967295, Default=0

ClearCacheDisabled: Clear cached secrets disabled.

VHDX: If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false

DeviceCount: Read-only count of Devices. Default=0

Locked: True if the Disk is currently locked, false otherwise. Default=false

PvsDiskInfo[]: If successful, the PvsDiskInfo object(s) are returned.

EXAMPLE 1: Get PvsDiskInfo for Farm

Get all PvsDiskInfo for the Farm.

```
Get-PvsDiskInfo
```

EXAMPLE 2: Get PvsDiskInfo for Site

Get all PvsDiskInfo for the Site named theSite.

```
Get-PvsDiskInfo -SiteName theSite
```

EXAMPLE 3: Get PvsDiskInfo for Site and Store

Get all PvsDiskInfo for the Site named theSite and Store named theStore.

```
Get-PvsDiskInfo -SiteName theSite -StoreName theStore
```

EXAMPLE 4: Get PvsDiskInfo for DiskLocator

Get the PvsDiskInfo for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
Get-PvsDiskInfo -Name theDiskLocator -SiteName theSite -StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 5: Get PvsDiskInfo and Enable

Get all PvsDiskInfo that are not Enabled and then Enables them.

```
Get-PvsDiskInfo -Fields Enabled | Where-Object {$_.Enabled -eq $false} | foreach { $o = $_; $o.Enabled = $true; $o } | Set-PvsDiskLocator
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Get-PvsDiskInventory

Get the fields for Inventory Status of a Disk Version or all Disk Versions for a Disk Locator.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Get Disk Version Inventory of.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator to Get Disk Version Inventory of.

Optional

uint Version: Specific Version to Get.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

(DiskLocatorId and Version) or DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Version: Version number. The base disk is version 0, the other version numbers are in part of the file name.

ServerId: GUID of the Server that the Disk Version Inventory is being reported about.

ServerName: Name of the Server that the Disk Version Inventory is being reported about.

FilePath: Path used to access the disk version from the Server. Empty if the information is not available.

FileTime: Date/Time of the date version file. Has the date and time without milliseconds. Empty if the information is not available.

PropertiesTime: Date/Time of the disk properties. Has the date and time without milliseconds. Empty if the information is not available.

State: The number code of the inventory state. Values are: 0 (Up to date), 1 (version file is missing), 2 (version file is out of date), 3 (properties are missing), 4 (properties are out of date), 5 (server is not reachable).

Active: 1 if the Server is currently active, 2 if unknown, and 0 otherwise.

PvsDiskInventory[]: If successful, the PvsDiskInventory object(s) are returned.

EXAMPLE 1: Get PvsDiskInventory for DiskLocator

Get all PvsDiskInventory for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

Get-PvsDiskInventory -Name theDiskLocator -SiteName theSite -StoreName theStore

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Get-PvsDiskLocator

Get the fields for a Disk Locator or all Disk Locators for a Device, Server, Store, Site, or Farm. All DiskLocators are returned if no parameters are passed.

One of these optional

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Get.

Guid[] DeviceId: GUID of the Device to Get all DiskLocators for.

string[] DeviceName: Name of the Device to Get all DiskLocators for.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Get all DiskLocators for.

Guid[] ServerId: GUID of the Server to Get all DiskLocators for.

string[] ServerName: Name of the Server to Get all DiskLocators for.

Guid[] UpdateTaskId: GUID of the Update Task to Get all DiskLocators for.

Guid[] SiteId: GUID of the Site to Get all DiskLocators for.

string[] SiteName: Name of the Site to Get all DiskLocators for.

or one of these optional & resolutions

string[] Name or DiskLocatorName: Name of the Disk Locator File to Get.

Guid[] StoreId: GUID of the Store to Get all DiskLocators for.

string[] StoreName: Name of the Store to Get all DiskLocators for.

string[] UpdateTaskName: Name of the Update Task to Get all DiskLocators for.

One of these optional

SwitchParameter Single: If -Single is specified, include single server connection. If this and All are not included, both connection types are included.

SwitchParameter All: If -All is specified, include all server connections for the store. If this and Single are not included, both connection types are included.

Optional

SwitchParameter OnlyActive: If -OnlyActive is specified, include only the active DiskLocators. If not included, all DiskLocators are returned.

SwitchParameter UpdateDevice: If -UpdateDevice is specified, include only DiskLocators that have an Update Device.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to Get all DiskLocators for.

string[] SiteName: Name of the Site to Get all DiskLocators for.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store to Get all DiskLocators for.

string[] StoreName: Name of the Store to Get all DiskLocators for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId, DeviceId, ServerId, UpdateTaskId, SiteId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or DiskLocatorId: Read-only GUID that uniquely identifies this Disk Locator.

Name or DiskLocatorName: Name of the Disk Locator File. It is unique within the Store. ASCII Max Length=52

SiteId: GUID of the Site this DiskLocator is to be a member of. It is not used with SiteName.

SiteName: Name of the Site this DiskLocator is to be a member of. It is not used with SiteId.

StoreId: GUID of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreName.

StoreName: Name of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreId.

Description: User description. Default="" Max Length=250

MenuText: Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default="" ASCII Max Length=64

ServerId: GUID of the single Server that this Disk Locator is assigned to. It is not used with ServerName. Default=00000000-0000-0000-0000-000000000000

ServerName: Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=""

Enabled: True when this disk can be booted, false otherwise. Default=true

Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, 300 is Collection Administrator, and 999 is read-only. Default=999

Mapped: True if the Disk is currently mapped, false otherwise. Default=false

EnabledForDevice: True when this disk is enabled for the Device specified, false otherwise. This is only returned when a Device is specified. Default=true

Active: True if the DiskLocator is currently active, false otherwise. Default=false

RebalanceEnabled: True when this Server can automatically rebalance Devices, false otherwise. Default=false

RebalanceTriggerPercent: Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25

SubnetAffinity: Qualifier for subnet affinity when assigning a Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2, Default=0

DiskUpdateDeviceId: GUID of the DiskUpdateDevice that is used when updates are performed. Default=00000000-0000-0000-0000-000000000000

DiskUpdateDeviceName: Name of the DiskUpdateDevice that is used when updates are performed. Default=""

TemporaryVersionSet: Read-only true when temporary version(s) are set. Default=false

PvsDiskLocator[]: If successful, the PvsDiskLocator object(s) are returned.

EXAMPLE 1: Get PvsDiskLocator for Farm

Get all PvsDiskLocator for the Farm.

```
Get-PvsDiskLocator
```

EXAMPLE 2: Get PvsDiskLocator for Site

Get all PvsDiskLocator for the Site named theSite.

```
Get-PvsDiskLocator -SiteName theSite
```

EXAMPLE 3: Get PvsDiskLocator for Site and Store

Get all PvsDiskLocator for the Site named theSite and Store named theStore.

```
Get-PvsDiskLocator -SiteName theSite -StoreName theStore
```

EXAMPLE 4: Get PvsDiskLocator for DiskLocator

Get the PvsDiskLocator for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 5: Get PvsDiskLocator and Enable

Get all PvsDiskLocator that are not Enabled and then Enables them.

```
Get-PvsDiskLocator -Fields Enabled | Where-Object {$_.Enabled -eq $false} | foreach { $o = $_; $o.Enabled = $true; $o } | Set-PvsDiskLocator
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Get-PvsDiskLocatorCount

Get count of Disk Locators for a Site and Type.

One of these required

Guid SiteId: GUID of the Site to get the Disk Locator Count of.

string SiteName: Name of the Site to get the Disk Locator Count of.

One of these optional

SwitchParameter Single: If -Single is specified, include single server connection. If this and All are not included, both connection types are included.

SwitchParameter All: If -All is specified, include all server connections for the store. If this and Single are not included, both connection types are included.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

SiteId

UInt32: If successful, the numeric value is returned

EXAMPLE 1: Get-PvsDiskLocatorCount Returns the Number (or Count) of PvsDiskLocator in PvsSite

```
Get-PvsDiskLocatorCount -SiteName theSite
```

Get-PvsDiskLocatorLock

Get the fields for all the locks of a Disk Locator.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Get the Locks.

or this required & resolution

string[] Name or DiskLocatorName: Name of Disk Locator to Get the Locks.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Exclusive: True when the lock is exclusive, false when it is shared. Default=false

DeviceId: GUID of the Device that has the lock, will be 00000000-0000-0000-0000-000000000000 if a Server has the lock.

DeviceName: Name of the Device that has the lock, will not be included if a Server has the lock.

ServerId: GUID of the Server that has the lock, will be 00000000-0000-0000-0000-000000000000 if a Device has the lock.

ServerName: Name of the Server that has the lock, will not be included if a Device has the lock.

ReadOnly: True when lock is because file system is read only, false when file system is read write Default=false

PvsDiskLocatorLock[]: If successful, the PvsDiskLocatorLock object(s) are returned.

EXAMPLE 1: Get PvsDiskLocatorLock for DiskLocator

Get all PvsDiskLocatorLock for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
Get-PvsDiskLocatorLock -Name theDiskLocator -SiteName theSite -
StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Get-PvsDiskUpdateDevice

Get the fields and status for a Disk Update Device, or all Disk Update Devices for a Site, Server, DiskLocator or Farm. All Disk Update Devices are returned if no parameters are passed.

One of these optional

Guid[] Guid or DeviceId: GUID of the Disk Update Device to Get.

string[] Name or DeviceName: Name of Disk Update Device to Get.

PvsPhysicalAddress[] DeviceMac: MAC of the Disk Update Device to Get.

Guid[] ServerId: GUID of the Server to Get all Disk Update Devices for.

string[] ServerName: Name of the Server to Get all Disk Update Devices for.

Guid[] DiskLocatorId: GUID of the DiskLocator to Get the Disk Update Device for.

Guid[] UpdateTaskId: GUID of the Update Task to Get all Disk Update Devices for.

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

or one of these optional & resolutions

string[] DiskLocatorName: Name of the DiskLocator to Get the Disk Update Device for.

string[] UpdateTaskName: Name of the Update Task to Get all Disk Update Devices for.

Optional

SwitchParameter OnlyActive: If -OnlyActive is specified, include only the active Disk Update Devices. Only active Disk Update Devices are always returned for ServerId or ServerName.

uint MakLicenseActivated: Optional MAK licensing indicator value to only return active Disk Update Devices for. Values are: 0 (MAK not used), 1 (Not Activated), 2 (Activated).

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId, ServerId, DiskLocatorId, UpdateTaskId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or DeviceId: Read-only GUID that uniquely identifies this Device.

Name or DeviceName: Computer name with no spaces. ASCII computer name characters Max Length=15

VirtualHostingPoolId: GUID of the Virtual Hosting Pool. It is not used with VirtualHostingPoolName. Default=00000000-0000-0000-0000-000000000000

VirtualHostingPoolName: Name of the Virtual Hosting Pool.

DiskLocatorId: GUID of the Disk Locator to update with this Device.

DiskLocatorName: Name of the Disk Locator File to update with this Device.

SiteId: GUID of the Site this Disk Update Device is to be a member of.

SiteName: Name of the Site this Disk Update Device is to be a member of.

StoreId: GUID of the Store that the Disk Locator is a member of.

StoreName: Name of the Store that the Disk Locator is a member of.

Description: User description. Default="" Max Length=250

DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device.

Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

Active: True if the Device is currently active, false otherwise. Default=false

AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" Max Length=256

Ip: Read-only IP of the Device. It is equal to 0.0.0.0 if the Device is not active.

ServerPortConnection: Read-only Port of the Server that the Device is using. It is equal to 0 if the Device is not active. Default=0

ServerIpConnection: Read-only IP of the Server that the Device is using. It is equal to 0.0.0.0 if the Device is not active.

ServerId: Read-only GUID of the Server that the Device is using. It is equal to 00000000-0000-0000-0000-000000000000 if the Device is not active.

ServerName: Read-only Name of the Server that the Device is using. It is equal to "" if the Device is not active.

DiskVersion: Read-only version of the Disk Locator File that the Device is using. It is equal to 0 if the Device is not active. Default=0

Status: 1 or 2 numbers in the format n,n. They are the number of retries and if ram cache is being used, ram cache percent used. It is equal to "" if the Device is not active.

LicenseType: 0 when None, 1 for Desktop, 2 for Server, 5 for OEM SmartClient, 6 for XenApp, 7 for XenDesktop. It is equal to 0 if the Device is not active. Default=0

MakLicenseActivated: Read-only indicator if MAK licensing is being used and is activated. Values are: 0 (MAK not used), 1 (Not Activated), 2 (Activated). It is equal to 0 if the Device is not active. Default=0

Model: Oem Only: Read-only model of the computer. Values are OptiPlex 745, 755, 320, 760, FX160, or Default. It is equal to "" if the Device is not active.

License: Oem Only: Read-only type of the license. Values are 0 when None, 1 or 2 when Desktop. It is equal to 0 if the Device is not active. Default=0

PvsDiskUpdateDevice[]: If successful, the PvsDiskUpdateDevice object(s) are returned.

EXAMPLE 1: Get PvsDiskUpdateDevice for Farm

Get all PvsDiskUpdateDevice for the Farm.

```
Get-PvsDiskUpdateDevice
```

EXAMPLE 2: Get PvsDiskUpdateDevice for Site

Get all PvsDiskUpdateDevice for the Site named theSite.

```
Get-PvsDiskUpdateDevice -SiteName theSite
```

EXAMPLE 3: Get PvsDiskUpdateDevice for Server

Get all PvsDiskUpdateDevice for the Server named theServer.

```
Get-PvsDiskUpdateDevice -ServerName theServer
```

EXAMPLE 4: Get PvsDiskUpdateDevice for DiskLocator

Get all PvsDiskUpdateDevice for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
Get-PvsDiskUpdateDevice -DiskLocatorName theDiskLocator -SiteName theSite -StoreName theStore
```

DiskLocatorId can be used instead of DiskLocatorName so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 5: Get PvsDiskUpdateDevice for Device

Get the PvsDiskUpdateDevice for the Device named theDevice.

```
Get-PvsDiskUpdateDevice -Name theDevice
```

EXAMPLE 6: Get PvsDiskUpdateDevice for Device MAC

```
Get the PvsDiskUpdateDevice for the Device with MAC 02-50-F2-00-00-01.

Get-PvsDiskUpdateDevice -DeviceMac "02-50-F2-00-00-01"
```

EXAMPLE 7: Get PvsDiskUpdateDevice for UpdateTask

```
Get the PvsDiskUpdateDevice for the UpdateTask named theUpdateTask in
                the Site named theSite.

Get-PvsDiskUpdateDevice -UpdateTaskName theUpdateTask -SiteName
                theSite

UpdateTaskId can be used instead of UpdateTaskName so that the
                SiteName or SiteId is not also needed.
```

Get-PvsDiskUpdateStatus

```
Get the status of an Update Task, or all Update Tasks for a Site or Farm.
  All Disk Update Tasks are returned if no parameters are passed.

  One of these optional

    Guid[] UpdateTaskId: GUID of the Update Task to Get.

    Guid[] DeviceId: GUID of the Disk Update Device to Get Disk Update
                Status for.

    string[] DeviceName: Name of the Disk Update Device to Get Disk Update
                Status for.

    PvsPhysicalAddress[] DeviceMac: MAC of the Disk Update Device to Get
                Disk Update Status for.

    Guid[] Guid or DiskUpdateTaskId: GUID of the Disk Update Task and
                Device relationship to Get Disk Update Status for.

    Guid[] SiteId: GUID of the Site to Get all Update Tasks for. Also used
                with UpdateTaskName.

    string[] SiteName: Name of the Site to Get all Update Tasks for. Also
                used with UpdateTaskName.

  or this optional & resolution

    string[] UpdateTaskName: Name of the Update Task to Get.

  One of these resolutions when needed

    Guid[] SiteId: GUID of the Site to Get all Update Tasks for. Also used
                with UpdateTaskName.

    string[] SiteName: Name of the Site to Get all Update Tasks for. Also
                used with UpdateTaskName.

  Instead of a parameter that matches one of the members listed

    PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      UpdateTaskId, DeviceId or SiteId

  If only selected fields are needed, pass them in the Fields parameter
    as a string array.
```

Guid or DiskUpdateTaskId: GUID that uniquely identifies this Update Task and Device relationship.

UpdateTaskId: GUID that uniquely identifies the Update Task.

UpdateTaskName: Name of the Update Task.

Description: User description of the Update Task.

DiskLocatorId: GUID of the Disk Locator to update.

Name or DiskLocatorName: Name of the Disk Locator File to update.

VirtualHostingPoolId: GUID of the Virtual Hosting Pool being used for the update.

VirtualHostingPoolName: Name of the Virtual Hosting Pool being used for the update.

DeviceId: GUID that Device being used to do the update.

DeviceName: Name of the Device being used to do the update.

SiteId: GUID of the Site that this Update Task Name is a member of.

SiteName: Name of the Site that this Update Task Name is a member of.

StoreId: GUID of the Store that the Disk Locator is a member of.

StoreName: Name of the Store that the Disk Locator is a member of.

PreviousResult: Status of the last run. Values are: 0 (Ready), 1 (Update Pending), 2 (Preparing Image), 3 (Starting VM), 4 (Update In Progress), 5 (Stopping VM), 6 (Submitting Image), 7 (Reverting Image), 8 (Invalid), 9 (Aborted), 10 (Completed Successfully), 11 (No Updates) Min=0, Max=11, Default=0

PreviousResultMessage: Message string that includes the results of the last run. Default="" Max Length=255

CurrentStatus: Current status of the update. Values are: 0 (Ready), 1 (Update Pending), 2 (Preparing Image), 3 (Starting VM), 4 (Update In Progress), 5 (Stopping VM), 6 (Submitting Image), 7 (Reverting Image), 8 (Invalid), 9 (Aborted), 10 (Completed Successfully), 11 (No Updates) Min=0, Max=11, Default=0

CurrentStatusMessage: Message string that includes the results of the run. Default="" Max Length=255

PvsDiskUpdateStatus[]: If successful, the PvsDiskUpdateStatus object(s) are returned.

EXAMPLE 1: Get PvsDiskUpdateStatus for Farm

Get all PvsDiskUpdateStatus for the Farm.

```
Get-PvsDiskUpdateStatus
```

EXAMPLE 2: Get PvsDiskUpdateStatus for Site

Get all PvsDiskUpdateStatus for the Site named theSite.

```
Get-PvsDiskUpdateStatus -SiteName theSite
```

EXAMPLE 3: Get PvsDiskUpdateStatus for Server

```
Get all PvsDiskUpdateStatus for the Server named theServer.

Get-PvsDiskUpdateStatus -ServerName theServer
```

EXAMPLE 4: Get PvsDiskUpdateStatus for DiskLocator

```
Get all PvsDiskUpdateStatus for the DiskLocator named theDiskLocator
                in the Site named theSite and Store named theStore.

Get-PvsDiskUpdateStatus -DiskLocatorName theDiskLocator -SiteName
                theSite -StoreName theStore

DiskLocatorId can be used instead of DiskLocatorName so that the
                SiteName or SiteId and StoreName or StoreId are not
                also needed.
```

EXAMPLE 5: Get PvsDiskUpdateStatus for Device

```
Get the PvsDiskUpdateStatus for the Device named theDevice.

Get-PvsDiskUpdateStatus -Name theDevice
```

EXAMPLE 6: Get PvsDiskUpdateStatus for Device MAC

```
Get the PvsDiskUpdateStatus for the Device with MAC 02-50-F2-00-00-01.

Get-PvsDiskUpdateStatus -DeviceMac "02-50-F2-00-00-01"
```

EXAMPLE 7: Get PvsDiskUpdateStatus for UpdateTask

```
Get the PvsDiskUpdateStatus for the UpdateTask named theUpdateTask in
                the Site named theSite.

Get-PvsDiskUpdateStatus -UpdateTaskName theUpdateTask -SiteName
                theSite

UpdateTaskId can be used instead of UpdateTaskName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 8: Get PvsDiskUpdateStatus for DiskUpdateTaskId

```
Get the PvsDiskUpdateStatus for the Device named theDevice then uses
                the DiskUpdateTaskId of the PvsDiskUpdateStatus to
                get the future Get-PvsDiskUpdateStatus.

$u = Get-PvsDiskUpdateStatus -DeviceName theDevice -Fields
                DiskUpdateTaskId

Get-PvsDiskUpdateStatus -Guid $u.DiskUpdateTaskId

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

Get-PvsDiskVersion

```
Get the fields for a Disk Version or all Disk Versions for a Disk Locator.

    This required

     Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Get Disk
                Versions of.

    or this required & resolution
```

string[] Name or DiskLocatorName: Name of the Disk Locator to Get Disk Versions of.

One of these optional

uint Version: Specific Version to Get.

uint Type: When set to 1, get the Maintenance or MaintenanceHighestVersion access version if it exists. When set to 2, get the Test access versions if any exist. When set to 3, get the Override access version if it exists.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

(DiskLocatorId and Version) or DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Version: Read-only version number. The base disk is version 0, the other version numbers are in part of the file name. Default=0

Description: User description. Default="" Max Length=250

Type: Read-only type of the Disk Version. Values are: 0 (Base), 1 (Manual), 2 (Automatic), 3 (Merge), and 4 (MergeBase) Min=0, Max=4, Default=0

CreateDate: Read-only Date/Time that the Disk Version was created. Default=getdate

ScheduledDate: Date/Time that the Disk Version is scheduled to become available. Has the date, hour and minute. Empty when the disk version is made available immediately. Default=Empty

DeleteWhenFree: Read-only true if the Disk Version is no longer needed because of a merge. If not current booted by a Device, it can be deleted. Default=false

Access: Read-only access of the Disk Version. Values are: 0 (Production), 1 (Maintenance), 2 (MaintenanceHighestVersion), 3 (Override), 4 (Merge), 5 (MergeMaintenance), 6 (MergeTest), and 7 (Test) Min=0, Max=7, Default=0

Name or DiskFileName: Name of the Disk File including the extension. Default=""

DeviceCount: Read-only count of Devices. Default=0

GoodInventoryStatus: True when the up to date file is accessible by all Servers, false otherwise. Default=false

TaskId: When a Merge is occurring, this will be set with the task number of the process that is occurring. Default=""

CanDelete: Read-only true when the version can be deleted. Default=false

CanMerge: Read-only true when the version can be update merged. Will be set for the highest version number. Default=false

CanMergeBase: Read-only true when the version can be base merged. Will be set for the highest version number. Default=false

CanPromote: Read-only true when the version can be promoted. Default=false

CanRevertTest: Read-only true when the version can be reverted to Test Access. Default=false

CanRevertMaintenance: Read-only true when the version can be reverted to Maintenance Access. Default=false

CanSetScheduledDate: Read-only true when the version can have the scheduled date modified. Default=false

CanOverride: Read-only true when the version can be set as the Override. Default=false

IsPending: Read-only true when the version ScheduledDate has not occurred. Default=false

TemporaryVersionSet: Read-only true when temporary version(s) are set. Some changes cannot be made to the version when this is set. Default=false

PvsDiskVersion[]: If successful, the PvsDiskVersion object(s) are returned.

EXAMPLE 1: Get PvsDiskVersion for DiskLocator

Get all PvsDiskVersion for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
Get-PvsDiskVersion -Name theDiskLocator -SiteName theSite -StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 2: Get PvsDiskVersion for DiskLocator

Get the first base PvsDiskVersion for the DiskLocator named theDiskLocator in Site named theSite and Store named theStore.

```
Get-PvsDiskVersion -Name theDiskLocator -Version 0 -SiteName theSite -StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 3: Get Maintenance PvsDiskVersion for DiskLocator

```
Get the Maintenance PvsDiskVersion for the DiskLocator named
                theDiskLocator in Site named theSite and Store named
                theStore.

Get-PvsDiskVersion -Name theDiskLocator -Type 1 -SiteName theSite -
                StoreName theStore

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 4: Get Test PvsDiskVersion for DiskLocator

```
Get all Test PvsDiskVersion for the DiskLocator named theDiskLocator
                in Site named theSite and Store named theStore.

Get-PvsDiskVersion -Name theDiskLocator -Type 2 -SiteName theSite -
                StoreName theStore

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 5: Get Override PvsDiskVersion for DiskLocator

```
Get the Override PvsDiskVersion for the DiskLocator named
                theDiskLocator in Site named theSite and Store named
                theStore.

Get-PvsDiskVersion -Name theDiskLocator -Type 3 -SiteName theSite -
                StoreName theStore

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

Get-PvsExists

```
Return true if a Site, Server, Collection, View, Device, Store, Update Task
   or Virtual Hosting Pool Name is already used. If a CollectionName,
   SiteViewName, UpdateTaskName or VirtualHostingPoolName is specified,
   SiteName or SiteId must be included.

    One of these required

     string SiteName: Name of the Site.

     string ServerName: Server name, to see if it is already used in the
                Farm.

     string FarmViewName: Farm View name, to see if it is already used in
                the Farm.

     string DeviceName: Device name, to see if it is already used in the
                Farm.

     PvsPhysicalAddress[] DeviceMac: Device MAC, to see if it is already
                used in the Farm.

     string StoreName: Store name, to see if it is already used.

    or one of these required & resolutions

     string CollectionName: Collection name, to see if it is already used
                in a Site.
```

string SiteViewName: Site View name, to see if it is already used in the Site.

string VirtualHostingPoolName: Virtual Hosting Pool name, to see if it is already used in a Site.

string UpdateTaskName: Update Task name, to see if it is already used in a Site.

string DiskLocatorName: DiskLocator name, to see if it is already used in a Site.

One of these resolutions when needed

string SiteName: Name of the Site.

Guid SiteId: GUID of the Site.

One of these resolutions when needed

Guid StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string StoreName: Store name, to see if it is already used.

Boolean: If successful, $true or $false is returned.

EXAMPLE 1: Get-PvsExists Determine if SiteName Already Exists

```
Get-PvsExists -SiteName theName
```

EXAMPLE 2: Get-PvsExists Determine if ServerName Already Exists

```
Get-PvsExists -ServerName theName
```

EXAMPLE 3: Get-PvsExists Determine if FarmViewName Already Exists

```
Get-PvsExists -FarmViewName theName
```

EXAMPLE 4: Get-PvsExists Determine if DeviceName Already Exists

```
Get-PvsExists -DeviceName theName
```

EXAMPLE 5: Get-PvsExists Determine if DeviceMac Already Exists

```
Get-PvsExists -DeviceMac "00-11-22-33-44-55"
```

EXAMPLE 6: Get-PvsExists Determine if StoreName Already Exists

```
Get-PvsExists -StoreName theName
```

EXAMPLE 7: Get-PvsExists Determine if CollectionName Already Exists

```
Get-PvsExists -CollectionName theName -SiteName theSite
```

SiteId can be used instead of SiteName.

EXAMPLE 8: Get-PvsExists Determine if SiteViewName Already Exists

```
Get-PvsExists -SiteViewName theName -SiteName theSite
```

SiteId can be used instead of SiteName.

EXAMPLE 9: Get-PvsExists Determine if VirtualHostingPoolName Already Exists

```
Get-PvsExists -VirtualHostingPoolName theName -SiteName theSite
```

SiteId can be used instead of SiteName.

EXAMPLE 10: Get-PvsExists Determine if UpdateTaskName Already Exists

```
Get-PvsExists -UpdateTaskName theName -SiteName theSite

SiteId can be used instead of SiteName.
```

EXAMPLE 11: Get-PvsExists Determine if DiskLocatorName Already Exists

```
Get-PvsExists -DiskLocatorName theName -SiteName theSite -StoreName
                theStore

SiteId can be used instead of SiteName, and StoreId can be used
                instead of StoreName.
```

Get-PvsFarm

```
Get the fields for the Farm.

    Optional

     Guid[] Guid or FarmId: GUID of the Farm to Get. This is optional since
                there is only one Farm.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      FarmId

    If only selected fields are needed, pass them in the Fields parameter
      as a string array.

     Guid or FarmId: Read-only GUID that uniquely identifies this Farm.

     Name or FarmName: Name of the Farm. Default="" Max Length=50

     Description: User description. Default="" Max Length=250

     AutoAddEnabled: True when Auto Add is enabled, false otherwise.
                Default=false

     AuditingEnabled: True when Auditing is enabled, false otherwise.
                Default=false

     LastAuditArchiveDate: Last date of Audit Trail data that was Archived.
                Has the date. Default=Empty

     DefaultSiteId: GUID of the Site to place new Devices into
                automatically. Not used with defaultSiteName.
                Default=00000000-0000-0000-0000-000000000000

     DefaultSiteName: Name of the Site to place new Devices into
                automatically. Not used with DefaultSiteId.
                Default=""

     OfflineDatabaseSupportEnabled: True when Offline Database Support is
                enabled, false otherwise. Default=false

     AdGroupsEnabled: Active Directory groups are used for authorization,
                when set to true. Windows groups are used when set to
                false. Default=false

     LicenseServer: License server name. Default="" Max Length=255

     LicenseServerPort: License server port. Min=1025, Max=65534,
                Default=27000
```

LicenseTradeUp: License server trade up, when set to true. Default=false

AutomaticMergeEnabled: True when Automatic Merge is enabled, false otherwise. If the number of versions becomes more than the MaxVersions value, a merge will occur at the end of PromoteDiskVersion. Default=true

MaxVersions: Maximum number a versions of a Disk that can exist before a merge will automatically occur. Min=3, Max=50, Default=5

MergeMode: Mode to place the version in after a merge has occurred. Values are: 0 (Production), 1 (Test) and 2 (Maintenance). Min=0, Max=2, Default=2

DatabaseServerName: Read-only name of the database server.

DatabaseInstanceName: Read-only name of the database instance.

DatabaseName: Read-only name of the database.

FailoverPartnerServerName: Read-only name of the database server.

FailoverPartnerInstanceName: Read-only name of the database server instance.

MultiSubnetFailover: Read-only Database MultiSubnetFailover value

Role: Read-only Role of the user for this item. 100 is Farm Administrator, and 999 is read-only. Default=999

PvsFarm[]: If successful, the PvsFarm object(s) are returned.

EXAMPLE 1: Get PvsFarm

Get the PvsFarm.

```
Get-PvsFarm
```

Get-PvsFarmView

Get the fields for a Farm View or all Farm Views in the Farm. All Farm Views are returned if no parameters are passed.

One of these optional

Guid[] Guid or FarmViewId: GUID of the Farm View to Get.

string[] Name or FarmViewName: Name of the Farm View to Get.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

FarmViewId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or FarmViewId: Read-only GUID that uniquely identifies this Farm View.

Name or FarmViewName: name of the Farm View. Max Length=50

Description: User description. Default="" Max Length=250

DeviceCount: Read-only count of Devices in this Farm View. Default=0

ActiveDeviceCount: Read-only count of active Devices in this Farm View. Default=0

MakActivateNeededCount: Read-only count of active Devices that need MAK activation in this Farm View. Default=0

PvsFarmView[]: If successful, the PvsFarmView object(s) are returned.

EXAMPLE 1: Get PvsFarmView for Farm

Get all PvsFarmView for the Farm.

```
Get-PvsFarmView
```

EXAMPLE 2: Get PvsFarmView

Get the PvsFarmView for the FarmView named theFarmView.

```
Get-PvsFarmView -Name theFarmView
```

Get-PvsGroup

Get the fields for the Groups for the user or the System.

Optional

string[] Domain: Domain of user (may be the name of the local computer).

string[] User: Name of user.

SwitchParameter AdGroupsEnabled: Get Active Directory groups, when set to true. Get Windows groups, when set to false. If not included, the Farm AdGroupsEnabled setting is used.

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name: Name of the Group.

Guid: GUID of the Active Directory group. 00000000-0000-0000-0000-000000000000 for Windows groups.

PvsGroup[]: If successful, the PvsGroup object(s) are returned.

EXAMPLE 1: Get PvsGroup for System

Get all PvsGroup for the System.

```
Get-PvsGroup
```

EXAMPLE 2: Get PvsGroup for User

Get all the PvsGroup in the Domain named theDomain for the User named theUser.

```
Get-PvsGroup -Domain theDomain -User theUser
```

EXAMPLE 3: Get Active Directory PvsGroup

Get all the Active Directory PvsGroup in the System.

```
Get-PvsGroup -AdGroupsEnabled
```

EXAMPLE 4: Get Windows Groups PvsGroup

Get all the Windows Groups PvsGroup in the System.

```
Get-PvsGroup -AdGroupsEnabled:$false
```

Get-PvsLocalServer

Return one record with the local server's NetBios name

PvsLocalServer[]: If successful, the PvsLocalServer object(s) are returned.

EXAMPLE 1: Get PvsLocalServer

Get the PvsLocalServer for the PVS SoapServer connected to.

```
Get-PvsLocalServer
```

Get-PvsMaintenanceVersionExists

Return true if the if the DiskLocator has a maintenance version, false otherwise.

This required

Guid Guid or DiskLocatorId: DiskLocator GUID, to see if it has a maintenance version.

or this required & resolution

string Name or DiskLocatorName: DiskLocator name, to see if it has a maintenance version.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

One of these resolutions when needed

Guid StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

Boolean: If successful, $true or $false is returned.

EXAMPLE 1: Get-PvsMaintenanceVersionExists for Name

```
Get-PvsMaintenanceVersionExists -Name theDiskLocator -SiteName theSite
-StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Get-PvsMinimumLastAutoAddDeviceNumber

Get the minimum that the Device Number of the last Auto Added Device can be.

This required

Guid CollectionId: GUID of the Collection to get the Minimum LastAutoAddDeviceNumber for.

or this required & resolution

string CollectionName: Name of the Collection to get the Minimum LastAutoAddDeviceNumber for.

Optional

string AutoAddPrefix: The string put before the Device Number for Auto Add.

string AutoAddSuffix: The string put after the Device Number for Auto Add.

uint AutoAddNumberLength: The maximum length of the Device Number for Auto Add. This length plus the AutoAddPrefix length plus the AutoAddSuffix length must be less than 16.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

CollectionId

UInt32: If successful, the numeric value is returned

EXAMPLE 1: Get-PvsMinimumLastAutoAddDeviceNumber

This example gets the highest auto-add number used so far for a PvsDevice that starts with the AutoAddPrefix and AutoAddSuffix of the PVsCollection.

```
Get-PvsMinimumLastAutoAddDeviceNumber -CollectionName theCollection -
SiteName theSite
```

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Get-PvsMinimumLastAutoAddDeviceNumber for AutoAddPrefix with AutoAddSuffix and AutoAddNumberLength

This example gets the highest auto-add number used so far for a PvsDevice with name length up to 10 characters and starts with the AutoAddPrefix and AutoAddSuffix specified.

```
Get-PvsMinimumLastAutoAddDeviceNumber -CollectionName theCollection -
SiteName theSite -AutoAddPrefix "P" -AutoAddSuffix
"S" -AutoAddNumberLength 10
```

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

Get-PvsMountedDisk

Get the mounted disk, if there is one.

One of these optional

Guid ServerId: GUID of the Server.

string ServerName: Name of the Server.

One of these optional

Guid StoreId: GUID of the Store.

string StoreName: Name of the Store.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId or StoreId

PvsDiskLocator: If successful, the mapped PvsDiskLocator is returned.

EXAMPLE 1: Get-PvsMountedDisk

This example gets the PvsMappedDisk. If no disk is mapped, null is returned.

```
$thePvsDiskLocator = Get-PvsMappedDisk
if ($thePvsDiskLocator -ne $null)
{
     $thePvsDiskLocator.Name    # display the name of the mapped PvsDiskLocator
}
```

EXAMPLE 2: Get-PvsMountedDisk for ServerName

This example gets any mapped disk for the ServerName. If no disk is mapped, null is returned.

```
$thePvsDiskLocator = Get-PvsMappedDisk -ServerName theServer
if ($thePvsDiskLocator -ne $null)
{
     $thePvsDiskLocator.Name    # display the name of the mapped PvsDiskLocator
}
```

EXAMPLE 3: Get-PvsMountedDisk for StoreName

This example gets any mapped disk for the StoreName. If no disk is mapped, null is returned.

```
$thePvsDiskLocator = Get-PvsMappedDisk -StoreName theStore
if ($thePvsDiskLocator -ne $null)
```

```
{
     $thePvsDiskLocator.Name    # display the name of the mapped
                     PvsDiskLocator
}
```

Get-PvsMountedDriveLetter

If there is currently a Mounted Drive, return the Letter of the Drive.

String: If successful, the String value is returned.

EXAMPLE 1: Get-PvsMountedDriveLetter

```
This example gets any mapped disk drive letter. If no disk is mapped,
                a string with length greater than 0 is returned.
$theDriveLetter = Get-PvsMappedDisk
if ($theDriveLetter -ne $null -and $theDriveLetter.length -gt 0)
{
     $theDriveLetter    # display the drive letter
}
```

Get-PvsNewDiskVersion

Get new Disk versions for the Store on the Server specified.

One of these required

Guid[] ServerId: GUID of the Server to look for new Disk versions.

string[] ServerName: Name of the Server to look for new Disk versions.

One of these required

Guid[] StoreId: GUID of the Store that the Server services to look for new Disk versions.

string[] StoreName: Name of the Store that the Server services to look for new Disk versions.

Optional

SwitchParameter AutoAddEnabled: If -AutoAddEnabled is specified, undefined Disk versions found will be automatically added.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name: Name of the disk file without the extension.

Status: Status of the disk file. Values are: 0 (Valid), 1 (Missing Properties File), 2 (Access Denied), 3 (Access Denied and Missing Properties File), 4 (Invalid Disk File), 5 (Manifest Invalid)

PvsNewDiskVersion[]: If successful, the PvsNewDiskVersion object(s) are returned.

EXAMPLE 1: Get PvsNewDiskVersion

Get all PvsNewDiskVersion in the Store named theStore using the Server named theServer.

```
Get-PvsNewDiskVersion -ServerName theServer -StoreName theStore
```

EXAMPLE 2: Get PvsNewDiskVersion with AutoAdd

Get all PvsNewDiskVersion and Auto Adds them in the Store named theStore using the Server named theServer.

```
Get-PvsNewDiskVersion -ServerName theServer -StoreName theStore -AutoAddEnabled
```

Get-PvsServer

Get the fields for a Server, all Servers in a Site that use a Store, service a DiskLocator, or for the whole Farm. All Servers are returned if no parameters are passed.

One of these optional

Guid[] Guid or ServerId: GUID of the Server to Get.

string[] Name or ServerName: Name of the Server to Get.

Guid[] SiteId: GUID of the Site to Get all Servers.

string[] SiteName: Name of the Site to Get all Servers.

Guid[] DiskLocatorId: GUID of the Disk Locator to Get all Servers.

or this optional & resolution

string[] DiskLocatorName: Name of the Disk Locator File to Get all Servers.

One of these optional & resolutions

Guid[] StoreId: GUID of the Store to Get all Servers.

string[] StoreName: Name of the Store to Get all Servers.

Optional

SwitchParameter All: If -All is specified, with StoreId or StoreName all Servers for the Store including ones with invalid paths will be returned.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to Get all Servers.

string[] SiteName: Name of the Site to Get all Servers.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store to Get all Servers.

string[] StoreName: Name of the Store to Get all Servers.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId, SiteId, DiskLocatorId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or ServerId: Read-only GUID that uniquely identifies this Server.

Name or ServerName: Computer name with no spaces. ASCII computer name characters Max Length=21

SiteId: GUID of the Site this Server is to be a member of. It is not used with SiteName.

SiteName: Name of the Site this Server is to be a member of. It is not used with SiteId.

Description: User description. Default="" Max Length=250

AdMaxPasswordAge: Number of days before a password expires. Min=1, Max=30, Default=7

LicenseTimeout: Amount of seconds before a license times out. Min=15, Max=300, Default=30

VDiskCreatePacing: VDisk create time pacing in miliseconds. Min=0, Max=5, Default=0

FirstPort: Number of the first UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6910

LastPort: Number of the last UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6930

ThreadsPerPort: Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8

BuffersPerThread: Number of buffers per worker thread. Min=1, Max=128, Default=24

ServerCacheTimeout: Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8

IoBurstSize: Number of bytes read/writes can send in a burst of packets. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=4096, Max=61440, Default=32768

MaxTransmissionUnits: Ethernet maximum transmission unit size for the protocol for use for Server and Device. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=502, Max=16426, Default=1506

MaxBootDevicesAllowed: Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500

MaxBootSeconds: Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60

BootPauseSeconds: Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10

AdMaxPasswordAgeEnabled: Age the password, when set to true. Default=false

EventLoggingEnabled: Enable event logging, when set to true. Default=false

NonBlockingIoEnabled: Use non-Blocking IO, when set to true. Default=true

Role: Read-only Role of the user for this item. 100 is Farm Administrator, and 200 is Site Administrator. Default=999

Ip: One or more streaming ip addresses.

InitialQueryConnectionPoolSize: Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50

InitialTransactionConnectionPoolSize: Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50

MaxQueryConnectionPoolSize: Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000

MaxTransactionConnectionPoolSize: Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000

RefreshInterval: Interval, in number of seconds, the server should wait before refreshing settings. If set to 0, unused database connections are never released. Min=0, Max=32767, Default=300

UnusedDbConnectionTimeout: Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300

BusyDbConnectionRetryCount: Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2

BusyDbConnectionRetryInterval: Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25

LocalConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are local. A value of 0 disables the feature. Min=0, Max=128, Default=4

RemoteConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4

Active: 1 if the Server is currently active, 2 if unknown, and 0 otherwise. Min=0, Max=2, Default=0

LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=4

LogFileSizeMax: Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5

LogFileBackupCopiesMax: Maximum number of log file backups. Min=1, Max=50, Default=4

```
        PowerRating: A strictly relative rating of this Server's capabilities
                     when compared to other Servers in the Store(s) it
                     belongs too; can be used to help tune load balancing.
                     Min=0.1, Max=1000, Default=1

        ServerFqdn: Read-only fully qualified domain name. Default="" Max
                     Length=1024

        ManagementIp: IP address used for management communications between
                     Servers. Default=0.0.0.0

        LastCeipUploadAttempt: Time that this server last attempted a CEIP
                     upload. Default=Empty

        LastBugReportAttempt: Time that this server last attempted to upload
                     or generate a bug report bundle. Default=Empty

        LastBugReportStatus: Status of the last bug report on this server.
                     Default="" Max Length=250

        LastBugReportResult: Status of the last bug report on this server.
                     Default="" Max Length=4000

        LastBugReportSummary: Summary of the last bug report on this server.
                     Default="" Max Length=250

    PvsServer[]: If successful, the PvsServer object(s) are returned.
```

EXAMPLE 1: Get PvsServer for Farm

```
        Get all PvsServer for the Farm.

        Get-PvsServer
```

EXAMPLE 2: Get PvsServer for Site

```
        Get all PvsServer for the Site named theSite.

        Get-PvsServer -SiteName theSite
```

EXAMPLE 3: Get PvsServer for Site and Store

```
        Get all PvsServer for the Site named theSite and Store named theStore.

        Get-PvsServer -SiteName theSite -StoreName theStore
```

EXAMPLE 4: Get PvsServer for DiskLocator

```
        Get the PvsServer for the DiskLocator named theDiskLocator in the Site
                     named theSite and Store named theStore.

        Get-PvsServer -DiskLocatorName theDiskLocator -SiteName theSite -
                     StoreName theStore

        DiskLocatorId can be used instead of DiskLocatorName so that the
                     SiteName or SiteId and StoreName or StoreId are not
                     also needed.
```

EXAMPLE 5: Get PvsServer for Server

```
        Get the PvsServer for the Server named theServer.

        Get-PvsServer -Name theServer
```

EXAMPLE 6: Get PvsServer Not Active and Start

```
        Get all PvsServer that are not Active and then Start them.
```

```
Get-PvsServer -Fields Active | Where-Object {$_.Active -eq 0} | Start-
                PvsStreamService

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

Get-PvsServerBiosBootstrap

```
Oem Only: Get the bootstrap fields for the Server dell_bios.bin BIOS
  bootstrap file.

   One of these required

    Guid[] Guid or ServerId: GUID of the Server to Get the dell_bios.bin
                BIOS bootstrap file from.

    string[] Name or ServerName: Name of the Server to Get the
                dell_bios.bin BIOS bootstrap file from.

   Instead of a parameter that matches one of the members listed

    PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

     ServerId

   If only selected fields are needed, pass them in the Fields parameter
     as a string array.

    Enabled: Automatically update the BIOS on the target device with these
                setting when set to true, otherwise do not use these
                settings. Default=false

    DhcpEnabled: Use DHCP to retrieve target device IP when set to true,
                otherwise use the static domain, dnsIpAddresstrue and
                dnsIpAddress2 settings. Default=true

    Lookup: Use DNS to find the Server when set to true with the
                ServerName host value, otherwise use the
                bootservertrue_Ip, bootservertrue_Port,
                bootserver2_Ip, bootserver2_Port, bootserver3_Ip,
                bootserver3_Port, bootserver4_Ip, and
                bootserver4_Port settings. Default=true

    VerboseMode: Display verbose diagnostic information when set to true.
                Default=false

    InterruptSafeMode: Interrupt safe mode (use if target device hangs
                during boot) when set to true. Default=false

    PaeMode: PAE mode (use if PAE enabled in boot.ini of target device)
                when set to true. Default=false

    BootFromHdOnFail: For network recovery reboot to hard drive when set
                to true, restore network connection when set to
                false. Default=false

    RecoveryTime: When bootFromHdOnFail is 1, this is the number of
                seconds to wait before reboot to hard drive. Min=10,
                Max=60000, Default=50

    PollingTimeout: Login polling timeout in milliseconds. Min=1000,
                Max=60000, Default=5000
```

GeneralTimeout: Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000

Name or ServerName: Host to use for DNS lookup. Only used when Lookup is true. Default=IMAGESERVER1

Bootserver1_Ip: 1st boot server IP. Only used when Lookup is false.

Bootserver1_Port: 1st boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

Bootserver2_Ip: 2nd boot server IP. Only used when Lookup is false. Default=0.0.0.0

Bootserver2_Port: 2nd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

Bootserver3_Ip: 3rd boot server IP. Only used when Lookup is false. Default=0.0.0.0

Bootserver3_Port: 3rd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

Bootserver4_Ip: 4th boot server IP. Only used when Lookup is false. Default=0.0.0.0

Bootserver4_Port: 4th boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

Domain: Domain of the primary and secondary DNS servers. Only used when DhcpEnabled is false.

DnsIpAddress1: Primary DNS server IP. Only used when DhcpEnabled is false.

DnsIpAddress2: Secondary DNS server IP. Only used when DhcpEnabled is false.

PvsServerBiosBootstrap[]: If successful, the PvsServerBiosBootstrap object(s) are returned.

EXAMPLE 1: Get PvsServerBiosBootstrap for Server

Get all PvsServerBiosBootstrap for the Server named theServer.

```
Get-PvsServerBiosBootstrap -Name theServer
```

Get-PvsServerBootstrap

Get the bootstrap fields for the Server and named bootstrap file specified.

One of these required

Guid[] Guid or ServerId: GUID of the Server to Get the named bootstrap file from.

string[] ServerName: Name of the Server to Get the named bootstrap file from.

This required

string[] Name: Name of the bootstrap file.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

(ServerId and Name) or ServerId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Bootserver1_Ip: 1st boot server IP.

Bootserver1_Netmask: 1st boot server netmask. Default=0.0.0.0

Bootserver1_Gateway: 1st boot server gateway. Default=0.0.0.0

Bootserver1_Port: 1st boot server port. Min=1025, Max=65536, Default=6910

Bootserver2_Ip: 2nd boot server IP. Default=0.0.0.0

Bootserver2_Netmask: 2nd boot server netmask. Default=0.0.0.0

Bootserver2_Gateway: 2nd boot server gateway. Default=0.0.0.0

Bootserver2_Port: 2nd boot server port. Min=1025, Max=65536, Default=6910

Bootserver3_Ip: 3rd boot server IP. Default=0.0.0.0

Bootserver3_Netmask: 3rd boot server netmask. Default=0.0.0.0

Bootserver3_Gateway: 3rd boot server gateway. Default=0.0.0.0

Bootserver3_Port: 3rd boot server port. Min=1025, Max=65536, Default=6910

Bootserver4_Ip: 4th boot server IP. Default=0.0.0.0

Bootserver4_Netmask: 4th boot server netmask. Default=0.0.0.0

Bootserver4_Gateway: 4th boot server gateway. Default=0.0.0.0

Bootserver4_Port: 4th boot server port. Min=1025, Max=65536, Default=6910

VerboseMode: Display verbose diagnostic information when set to true. Default=false

InterruptSafeMode: Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false

PaeMode: PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false

BootFromHdOnFail: For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false

RecoveryTime: When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50

PollingTimeout: Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000

GeneralTimeout: Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000

PvsServerBootstrap[]: If successful, the PvsServerBootstrap object(s) are returned.

EXAMPLE 1: Get PvsServerBootstrap

Get the PvsServerBootstrap for the Bootstrap named theBootstrap on the Server named theServer.

```
Get-PvsServerBootstrap -ServerName theServer -Name theBootstrap
```

Get-PvsServerBootstrapName

Get the bootstrap names for a Server.

One of these required

Guid[] Guid or ServerId: GUID of the Server to Get bootstrap names for.

string[] Name or ServerName: Name of the Server to Get bootstrap names for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

PvsServerBootstrapName[]: If successful, the PvsServerBootstrapName object(s) are returned.

EXAMPLE 1: Get PvsServerBootstrapName for Server

Get all PvsServerBootstrapName for the Server named theServer.

```
Get-PvsServerBootstrapName -Name theServer
```

Get-PvsServerCount

Get count of Servers in a Site.

One of these required

Guid Guid or SiteId: GUID of the Site to get the Server Count of.

string Name or SiteName: Name of the Site to get the Server Count of.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

SiteId

UInt32: If successful, the numeric value is returned

EXAMPLE 1: Get-PvsServerCount Returns the Number (or Count) of PvsServer in PvsSite

```
Get-PvsServerCount -Name theSite
```

Get-PvsServerInfo

Get the fields and status for a Server, all Servers in a Site that use a Store, service a DiskLocator, or for the whole Farm. All Servers are returned if no parameters are passed.

One of these optional

Guid[] Guid or ServerId: GUID of the Server to Get.

string[] Name or ServerName: Name of the Server to Get.

Guid[] SiteId: GUID of the Site to Get all Servers.

string[] SiteName: Name of the Site to Get all Servers.

Guid[] DiskLocatorId: GUID of the Disk Locator to Get all Servers.

or this optional & resolution

string[] DiskLocatorName: Name of the Disk Locator File to Get all Servers.

One of these optional & resolutions

Guid[] StoreId: GUID of the Store to Get all Servers.

string[] StoreName: Name of the Store to Get all Servers.

Optional

SwitchParameter All: If -All is specified, with StoreId or StoreName all Servers for the Store including ones with invalid paths will be returned.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to Get all Servers.

string[] SiteName: Name of the Site to Get all Servers.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store to Get all Servers.

string[] StoreName: Name of the Store to Get all Servers.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId, SiteId, DiskLocatorId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or ServerId: Read-only GUID that uniquely identifies this Server.

Name or ServerName: Computer name with no spaces. ASCII computer name characters Max Length=21

SiteId: GUID of the Site this Server is to be a member of. It is not used with SiteName.

SiteName: Name of the Site this Server is to be a member of. It is not used with SiteId.

Description: User description. Default="" Max Length=250

AdMaxPasswordAge: Number of days before a password expires. Min=1, Max=30, Default=7

LicenseTimeout: Amount of seconds before a license times out. Min=15, Max=300, Default=30

VDiskCreatePacing: VDisk create time pacing in miliseconds. Min=0, Max=5, Default=0

FirstPort: Number of the first UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6910

LastPort: Number of the last UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6930

ThreadsPerPort: Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8

BuffersPerThread: Number of buffers per worker thread. Min=1, Max=128, Default=24

ServerCacheTimeout: Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8

IoBurstSize: Number of bytes read/writes can send in a burst of packets. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=4096, Max=61440, Default=32768

MaxTransmissionUnits: Ethernet maximum transmission unit size for the protocol for use for Server and Device. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=502, Max=16426, Default=1506

MaxBootDevicesAllowed: Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500

MaxBootSeconds: Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60

BootPauseSeconds: Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10

AdMaxPasswordAgeEnabled: Age the password, when set to true. Default=false

EventLoggingEnabled: Enable event logging, when set to true. Default=false

NonBlockingIoEnabled: Use non-Blocking IO, when set to true. Default=true

Role: Read-only Role of the user for this item. 100 is Farm Administrator, and 200 is Site Administrator. Default=999

Ip: One or more streaming ip addresses.

InitialQueryConnectionPoolSize: Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50

InitialTransactionConnectionPoolSize: Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50

MaxQueryConnectionPoolSize: Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000

MaxTransactionConnectionPoolSize: Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000

RefreshInterval: Interval, in number of seconds, the server should wait before refreshing settings. If set to 0, unused database connections are never released. Min=0, Max=32767, Default=300

UnusedDbConnectionTimeout: Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300

BusyDbConnectionRetryCount: Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2

BusyDbConnectionRetryInterval: Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25

LocalConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are local. A value of 0 disables the feature. Min=0, Max=128, Default=4

RemoteConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4

Active: 1 if the Server is currently active, 2 if unknown, and 0 otherwise. Min=0, Max=2, Default=0

LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=4

LogFileSizeMax: Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5

LogFileBackupCopiesMax: Maximum number of log file backups. Min=1, Max=50, Default=4

PowerRating: A strictly relative rating of this Server's capabilities when compared to other Servers in the Store(s) it belongs too; can be used to help tune load balancing. Min=0.1, Max=1000, Default=1

ServerFqdn: Read-only fully qualified domain name. Default="" Max Length=1024

ManagementIp: IP address used for management communications between Servers. Default=0.0.0.0

LastCeipUploadAttempt: Time that this server last attempted a CEIP upload. Default=Empty

LastBugReportAttempt: Time that this server last attempted to upload or generate a bug report bundle. Default=Empty

LastBugReportStatus: Status of the last bug report on this server. Default="" Max Length=250

LastBugReportResult: Status of the last bug report on this server. Default="" Max Length=4000

LastBugReportSummary: Summary of the last bug report on this server. Default="" Max Length=250

ContactIp: Read-only contact IP for the Server.

ContactPort: Read-only contact port for the Server.

DeviceCount: Read-only count of Devices. Default=0

PvsServerInfo[]: If successful, the PvsServerInfo object(s) are returned.

EXAMPLE 1: Get PvsServerInfo for Farm

Get all PvsServerInfo for the Farm.

```
Get-PvsServerInfo
```

EXAMPLE 2: Get PvsServerInfo for Site

Get all PvsServerInfo for the Site named theSite.

```
Get-PvsServerInfo -SiteName theSite
```

EXAMPLE 3: Get PvsServerInfo for Site and Store

Get all PvsServerInfo for the Site named theSite and Store named theStore.

```
Get-PvsServerInfo -SiteName theSite -StoreName theStore
```

EXAMPLE 4: Get PvsServerInfo for DiskLocator

Get the PvsServerInfo for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
Get-PvsServerInfo -DiskLocatorName theDiskLocator -SiteName theSite -StoreName theStore
```

DiskLocatorId can be used instead of DiskLocatorName so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 5: Get PvsServerInfo for Server

Get the PvsServerInfo for the Server named theServer.

```
Get-PvsServerInfo -Name theServer
```

EXAMPLE 6: Get PvsServerInfo Not Active and Start

Get all PvsServerInfo that are not Active and then Start them.

```
Get-PvsServerInfo -Fields Active | Where-Object {$_.Active -eq 0} | Start-PvsStreamService
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Get-PvsServerName

Return the name of the Server the SoapServer is running on.

String: If successful, the String value is returned.

EXAMPLE 1: Get-PvsServerName

```
Get-PvsServerName
```

Get-PvsServerStatus

Get the Server Status fields for a Server.

One of these required

Guid[] Guid or ServerId: GUID of the Server to Get status for.

string[] Name or ServerName: Name of Server to Get status for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or ServerId: Read-only GUID of the Server. Can be used with Get Server.

Name or ServerName: Read-only Name of the Server. Can be used with Get Server.

Ip: Read-only contact IP for the Server.

Port: Read-only contact port for the Server.

DeviceCount: Read-only count of Devices. Default=0

Status: Status of the server, 0 if down, 1 if up and 2 if unknown. Default=0

PvsServerStatus[]: If successful, the PvsServerStatus object(s) are returned.

EXAMPLE 1: Get PvsServerStatus for Server

Get the PvsServerStatus for the Server named theServer.

```
Get-PvsServerStatus -Name theServer
```

Get-PvsServerStore

Get the directory and cache paths of a Server for one or all Stores.

One of these required

Guid[] ServerId: GUID of a Server.

string[] ServerName: Name of a Server.

One of these optional

Guid[] StoreId: GUID of the Store.

string[] StoreName: Name of the Store.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

(ServerId and StoreId), ServerId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

StoreId: GUID of the Store. StoreName can be used instead.

StoreName: Name of the Store. StoreId can be used instead.

ServerId: GUID of the server that uses the Store. ServerName can be used instead.

ServerName: Name of the server that uses the Store. ServerId can be used instead.

Path: Directory path that the Server uses to access the Store. Default="" Max Length=255

CachePath: Cache path(s) that the Server uses with the Store. If none are specified the caches will be placed in the Store cachePath. Default=None

PvsServerStore[]: If successful, the PvsServerStore object(s) are returned.

EXAMPLE 1: Get PvsServerStore for Server

Get all PvsServerStore for the Server named theServer.

```
Get-PvsServerStore -ServerName theServer
```

EXAMPLE 2: Get PvsServerStore for Server and Store

Get the PvsServerStore for the Server named theServer and Store named theStore.

```
Get-PvsServerStore -ServerName theServer -StoreName theStore
```

Get-PvsServerStoreActiveDeviceCount

Get the count of Devices currently connected to any vdisk served from the Store by the Server.

One of these required

Guid ServerId: GUID of the Server.

string ServerName: Name of the Server.

One of these required

Guid StoreId: GUID of the Store.

string StoreName: Name of the Store.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId or StoreId

UInt32: If successful, the numeric value is returned

EXAMPLE 1: Get-PvsServerStoreActiveDeviceCount Returns the Number (or Count) of PvsDevice Served from PvsStore by PvsServer

```
Get-PvsServerStoreActiveDeviceCount -ServerName theServer -StoreName
    theStore
```

Get-PvsSite

```
Get the fields for a Site or all Sites. All Sites are returned if no
  parameters are passed.

  One of these optional

   Guid[] Guid or SiteId: GUID of the Site to Get.

   string[] Name or SiteName: Name of the Site to Get.

  Instead of a parameter that matches one of the members listed

   PvsObject[] Object: PvsObjects with the members below can be used as
                   the Object parameter or from a pileline:

    SiteId

  If only selected fields are needed, pass them in the Fields parameter
    as a string array.

   Guid or SiteId: Read-only GUID that uniquely identifies this Site.

   Name or SiteName: Name of the Site. Max Length=50

   Description: User description. Default="" Max Length=250

   DefaultCollectionId: GUID of the Collection to place new Devices into
                   automatically. Not used with defaultCollectionName.
                   Default=00000000-0000-0000-0000-000000000000

   DefaultCollectionName: Name of the Collection to place new Devices
                   into automatically. Not used with
                   DefaultCollectionId. Default=""

   InventoryFilePollingInterval: The number of seconds between polls for
                   Disk changes in the Stores. Min=1, Max=600,
                   Default=60

   EnableDiskUpdate: True when Disk Updated is enabled for the Site,
                   false otherwise. Default=false

   DiskUpdateServerId: GUID of the Disk Update Server for the Site. Not
                   used with DiskUpdateServerName. Default=00000000-
                   0000-0000-0000-000000000000

   DiskUpdateServerName: Name of the Disk Update Server for the Site. Not
                   used with DiskUpdateServerId. Default=""

   MakUser: User name used for MAK activation. Default="" Max Length=64

   MakPassword: User password used for MAK activation. Default="" Max
                   Length=64

   Role: Read-only Role of the user for this item. 100 is Farm
                   Administrator, 200 is Site Administrator, and 999 is
                   read-only. Default=999

  PvsSite[]: If successful, the PvsSite object(s) are returned.
```

EXAMPLE 1: Get PvsSite for Farm

```
Get all PvsSite for the Farm.

Get-PvsSite
```

EXAMPLE 2: Get PvsSite

```
Get the PvsSite for the Site named theSite.

Get-PvsSite -Name theSite
```

Get-PvsSiteView

```
Get the fields for a Site View or all Site Views in a Site or the whole
   Farm. All Site Views are returned if no parameters are passed.

    One of these optional

     Guid[] Guid or SiteViewId: GUID of the Site View to Get.

     Guid[] SiteId: GUID of the Site to Get all Views for.

     string[] SiteName: Name of the Site to Get all Views for.

    or this optional & resolution

     string[] Name or SiteViewName: Name of the Site View to Get.

    One of these resolutions when needed

     Guid[] SiteId: GUID of the Site to Get all Views for.

     string[] SiteName: Name of the Site to Get all Views for.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                         the Object parameter or from a pileline:

      SiteViewId or SiteId

    If only selected fields are needed, pass them in the Fields parameter
      as a string array.

     Guid or SiteViewId: Read-only GUID that uniquely identifies this Site
                         View.

     Name or SiteViewName: Name of the Site View. Max Length=50

     SiteId: GUID of the Site this View is to be a member of. It is not
                         used with SiteName.

     SiteName: Name of the Site this View is to be a member of. It is not
                         used with SiteId.

     Description: User description. Default="" Max Length=250

     DeviceCount: Read-only count of Devices in this Site View. Default=0

     DeviceWithPVDCount: Read-only count of Devices with Personal vDisk in
                         this Site View. Default=0

     ActiveDeviceCount: Read-only count of active Devices in this Site
                         View. Default=0

     MakActivateNeededCount: Read-only count of active Devices that need
                         MAK activation in this Site View. Default=0
```

Role: Read-only Role of the user for this item. 100 is Farm Administrator, and 200 is Site Administrator. Default=999

PvsSiteView[]: If successful, the PvsSiteView object(s) are returned.

EXAMPLE 1: Get PvsSiteView for Farm

Get all PvsSiteView for the Farm.

```
Get-PvsSiteView
```

EXAMPLE 2: Get PvsSiteView for Site

Get all PvsSiteView for the Site named theSite.

```
Get-PvsSiteView -SiteName theSite
```

EXAMPLE 3: Get PvsSiteView

Get the PvsSiteView for the SiteView named theSiteView in the Site named theSite.

```
Get-PvsSiteView -Name theSiteView -SiteName theSite
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Get-PvsStore

Get the fields for a Store or all Stores for a Server, Site or the Farm. All Stores are returned if no parameters are passed.

One of these optional

Guid[] Guid or StoreId: GUID of the Store to Get.

string[] Name or StoreName: Name of the Store to Get.

Guid[] ServerId: GUID of the Server to Get all Stores for.

string[] ServerName: Name of the Server to Get all Stores for.

Guid[] SiteId: GUID of the Site to Get all Stores for.

string[] SiteName: Name of the Site to Get all Stores for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

StoreId, ServerId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or StoreId: Read-only GUID that uniquely identifies this Store.

Name or StoreName: Name of the Store. Max Length=50

SiteId: GUID of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteName can be used instead. Default=00000000-0000-0000-0000-000000000000

SiteName: Name of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteId can be used instead. Default=""

Description: User description. Default="" Max Length=250

Path: Default directory path that the Servers use to access this Store. Max Length=255

CachePath: Default Cache path(s) that the Servers use with this Store. If none are specified the caches will be placed in the WriteCache subdirectory of the Store path. Default=None

Role: Read-only Role of the user for this item. 100 is Farm Administrator, 200 is Site Administrator, and 999 is read-only. Default=999

PathType: Read-only field indicating if the vdisks are on a server's local hard disk or on a remote share.

PvsStore[]: If successful, the PvsStore object(s) are returned.

EXAMPLE 1: Get PvsStore for Farm

Get all PvsStore for the Farm.

```
Get-PvsStore
```

EXAMPLE 2: Get PvsStore for Site

Get all PvsStore for the Site named theSite.

```
Get-PvsStore -SiteName theSite
```

EXAMPLE 3: Get PvsStore for Server

Get all PvsStore for the Server named theServer.

```
Get-PvsStore -ServerName theServer
```

EXAMPLE 4: Get PvsStore

Get the PvsStore for the Store named theStore.

```
Get-PvsStore -Name theStore -SiteName theSite
```

Get-PvsStoreFreeSpace

Get the free megabytes available in the Store.

One of these required & resolutions

Guid Guid or StoreId: GUID of the Store.

string Name or StoreName: Name of the Store.

One of these resolutions when needed

Guid ServerId: GUID of the Server to use to determine the free space in the Store.

string ServerName: Name of the Server to use to determine the free space in the Store.

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

UInt32: If successful, the numeric value is returned

EXAMPLE 1: Get-PvsStoreFreeSpace for Name with ServerName

```
Get-PvsStoreFreeSpace -Name theStore -ServerName theServer
```

EXAMPLE 2: Get-PvsStoreFreeSpace for Name with SiteName

```
Get-PvsStoreFreeSpace -Name theStore -SiteName theSite
```

Get-PvsStoreSharedOrServerPath

Get the Stores and paths for the ServerName specified or Stores with only shared UNC paths.

One of these required

Guid[] SiteId: GUID of a Site used for authorization check.

string[] SiteName: Name of a Site used for authorization check.

Optional

string[] ServerName: Name of a Server to also get local Store paths for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

StoreId: GUID of the Store.

StoreName: Name of the Store.

Path: Directory path that the Servers use to access this Store.

PvsStoreSharedOrServerPath[]: If successful, the PvsStoreSharedOrServerPath object(s) are returned.

EXAMPLE 1: Get PvsStoreSharedOrServerPath with Sharded UNC Paths

Get PvsStoreSharedOrServerPath with shared UNC paths for the Farm.

```
Get-PvsStoreSharedOrServerPath -SiteName theSite
```

EXAMPLE 2: Get PvsStoreSharedOrServerPath with Sharded UNC Paths and Server Local Paths

Get PvsStoreSharedOrServerPath with shared UNC paths for the Farm with local paths for the Server named theServer.

```
Get-PvsStoreSharedOrServerPath -SiteName theSite -StoreName theServer
```

Get-PvsTask

Get the current Task fields for select, or all active and completed un-cleared tasks.

One of these optional

uint TaskId: ID of the Task to get.

Guid[] SiteId: GUID of the Site to get Tasks for.

```
    string[] SiteName: Name of the Site to get Tasks for.

Optional

    uint State: The TaskState to get Tasks for. Values are: 0
                (Processing), 1 (Cancelled), and 2 (Complete).

Instead of a parameter that matches one of the members listed

    PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      TaskId or SiteId

If only selected fields are needed, pass them in the Fields parameter
  as a string array.

    TaskId: Unique ID of the task.

    SiteId: GUID of the Site that this Task is being processed in.
                Default=00000000-0000-0000-0000-000000000000

    SiteName: Name of the Site that that this Task is being processed in.

    Handle: Handle to a running function.

    ServerFqdn: Qualified name of the server. Default="" Max Length=1024

    Ip: IP Address of the remote host.

    Port: Port number of the remote service.

    StartTime: Time the task was started. Has the date and time without
                milliseconds.

    ExpirationTime: Time the task record may be removed from the database
                if the task does not complete. Has the date and time
                without milliseconds.

    State: State of the Task. Values are: 0 (Processing), 1 (Cancelled),
                and 2 (Complete). Min=0, Max=2

    CommandType: Type of the command. Values are: Add, Delete, Get, Info,
                Run, RunWithReturn, Set and SetList. Default="" Max
                Length=13

    Command: Command being processed. Default="" Max Length=50

    MapiException: Exception result in XML format. Default=""

    Results: Result in XML format. Default=""

PvsTask[]: If successful, the PvsTask object(s) are returned.
```

EXAMPLE 1: Get PvsTask for Farm

```
Get all PvsTask for the Farm.

Get-PvsTask
```

EXAMPLE 2: Get PvsTask for Site

```
Get all PvsTask for the Site named theSite.

Get-PvsTask -SiteName theSite
```

EXAMPLE 3: Get PvsTask Currently Processing

```
Get PvsTask that are currently in Processing state for the Farm.
```

```
Get-PvsTask -State 0
```

EXAMPLE 4: Get PvsTask

```
Get PvsTask for TaskId 101.
Get-PvsTask -TaskId 101
```

Get-PvsTaskStatus

```
Get the status of a Task in percent complete.
    This required
     uint TaskId: Id of the Task to get the Status of.
    Instead of a parameter that matches one of the members listed
     PvsObject[] Object: PvsObjects with the members below can be used as
                    the Object parameter or from a pileline:
      TaskId
    UInt32: If successful, the numeric value is returned
```

EXAMPLE 1: Get-PvsTaskStatus for Start-PvsBoot to PvsTask

```
$thePvsTask = Start-PvsBoot -Name theDevice                    # start
                    the task
while ($thePvsTask.State -eq 0)                                # while
                    the task is processing
{
     %percentFinished = Get-PvsTaskStatus -Object $thePvsTask   # get
                         percent finished
     %percentFinished.ToString() + "% finished"                 #
                         display percent finished
     Start-Sleep -seconds 10                                    # wait
                         10 seconds more
     $thePvsTask = Get-PvsTask -Object $thePvsTask              # get
                         the current PvsTask object
}
if ($thePvsTask.State -eq 2)                                   # check
                    to see if completed
{
     "Successful"
}
else
{
     "Cancelled"
}
```

EXAMPLE 2: Get-PvsTaskStatus for taskId

```
Get-PvsTaskStatus -taskId 101
```

Get-PvsUndefinedDisk

Get undefined Disks for the Store on the Server specified.

One of these required

Guid[] ServerId: GUID of the Server to look for undefined Disks.

string[] ServerName: Name of the Server to look for undefined Disks.

One of these required

Guid[] StoreId: GUID of the Store that the Server services to look for undefined Disks.

string[] StoreName: Name of the Store that the Server services to look for undefined Disks.

Optional

SwitchParameter AutoAddEnabled: If -AutoAddEnabled is specified, then undefined Disks found will be automatically added.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name: Name of the disk file without the extension.

Status: Status of the disk file. Values are: 0 (Valid), 1 (Missing Properties File), 2 (Access Denied), 3 (Access Denied and Missing Properties File), 4 (Invalid Disk File), 5 (Manifest Missing or Invalid), 6 (Both VHD and VHDX)

VHDX: If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false

PvsUndefinedDisk[]: If successful, the PvsUndefinedDisk object(s) are returned.

EXAMPLE 1: Get PvsUndefinedDisk

Get all PvsUndefinedDisk in the Store named theStore using the Server named theServer.

```
Get-PvsUndefinedDisk -ServerName theServer -StoreName theStore
```

EXAMPLE 2: Get PvsUndefinedDisk with AutoAdd

Get all PvsUndefinedDisk and Auto Adds them in the Store named theStore using the Server named theServer.

```
Get-PvsUndefinedDisk -ServerName theServer -StoreName theStore -
AutoAddEnabled
```

Get-PvsUpdateTask

Get the fields for an Update Task or all Update Tasks in a Site or the whole Farm. All Update Tasks are returned if no parameters are passed.

One of these optional

Guid[] Guid or UpdateTaskId: GUID of the Update Task to Get.

Guid[] SiteId: GUID of the Site to Get all Update Tasks for.

string[] SiteName: Name of the Site to Get all Update Tasks for.

or this optional & resolution

string[] Name or UpdateTaskName: Name of the Update Task to Get.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to Get all Update Tasks for.

string[] SiteName: Name of the Site to Get all Update Tasks for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

UpdateTaskId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or UpdateTaskId: Read-only GUID that uniquely identifies this Update Task.

Name or UpdateTaskName: Name of the Update Task. It is unique within the Site. Max Length=50

SiteId: GUID of the Site that this Update Task is a member of. It is not used with SiteName.

SiteName: Name of the Site that this Update Task is a member of. It is not used with SiteId.

Description: User description. Default="" Max Length=250

Enabled: True when it will be processed, false otherwise. Default=true

Hour: The hour of the day to perform the task. Min=0, Max=23, Default=0

Minute: The minute of the hour to perform the task. Min=0, Max=59, Default=0

Recurrence: The update will reoccur on this schedule. 0 = None, 1 = Daily, 2 = Every Weekday, 3 = Weekly, 4 = Monthly Date, 5 = Monthly Type. Min=0, Max=5, Default=0

DayMask: Days selected values. 1 = Monday, 2 = Tuesday, 4 = Wednesday, 8 = Thursday, 16 = Friday, 32 = Saturday, 64 = Sunday, 128 = Day. Default=0. This is used with Weekly and Monthly Type recurrence. Min=1, Max=255, Default=4

Date: Days of the month. Numbers from 1-31 are the only valid values. This is used with Monthly Date recurrence. Default="" Max Length=83

MonthlyOffset: When to happen monthly. 0 = None, 1 = First, 2 = Second, 3 = Third, 4 = Forth, 5 = Last. This is used with Monthly Type recurrence. Min=0, Max=5, Default=3

```
    EsdType: Esd to use. Valid values are SCCM or WSUS. If no value, a
                 custom script is run on the client. Default="" Max
                 Length=50

    PreUpdateScript: Script file to run before the update starts.
                 Default="" Max Length=255

    PreVmScript: Script file to run before the VM is loaded. Default=""
                 Max Length=255

    PostUpdateScript: Script file to run after the update finishes.
                 Default="" Max Length=255

    PostVmScript: Script file to run after the VM is unloaded. Default=""
                 Max Length=255

    Domain: Domain to add the Disk Update Device(s) to. If not included,
                 the first Domain Controller found on the Server is
                 used. Default="" Max Length=255

    OrganizationUnit: Organizational Unit to add the Disk Update Device(s)
                 to. This parameter is optional. If it is not
                 specified, the device is added to the built in
                 Computers container. Child OU's should be delimited
                 with forward slashes, e.g. "ParentOU/ChildOU".
                 Special characters in an OU name, such as '"', '#',
                 '+', ',', ';', '>', '=', must be escaped with a
                 backslash. For example, an OU called
                 "commaIn,TheMiddle" must be specified as
                 "commaIn\,TheMiddle". The old syntax of delimiting
                 child OU's with a comma is still supported, but
                 deprecated. Note that in this case, the child OU
                 comes first, e.g. "ChildOU,ParentOU". Default="" Max
                 Length=255

    PostUpdateApprove: Access to place the version in after the update has
                 occurred. 0 = Production, 1 = Test, 2 = Maintenance.
                 Min=0, Max=2, Default=0

  PvsUpdateTask[]: If successful, the PvsUpdateTask object(s) are
    returned.
```

EXAMPLE 1: Get PvsUpdateTask for Farm

```
    Get all PvsUpdateTask for the Farm.

    Get-PvsUpdateTask
```

EXAMPLE 2: Get PvsUpdateTask for Site

```
    Get all PvsUpdateTask for the Site named theSite.

    Get-PvsUpdateTask -SiteName theSite
```

EXAMPLE 3: Get PvsUpdateTask

```
    Get PvsUpdateTask for UpdateTask named theUpdateTask in Site named
                 theSite.

    Get-PvsUpdateTask -Name theUpdateTask -SiteName theSite

    Guid can be used instead of Name so that the SiteName or SiteId is not
                 also needed.
```

Get-PvsUploadCeip

Perform a one time upload of CEIP data. Return upload Id if successful.

This optional

string OneTimeUpload: If -OneTimeUpload is specified, perform a one time upload.

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

String: If successful, the String value is returned.

Get-PvsVersion

Get the version information.

If only selected fields are needed, pass them in the Fields parameter as a string array.

MapiVersion: Version of the system in major.minor.point.build format.

DbVersion: Version of the database schema as a number. Default=0

Type: Type of system. Values are 0 (Normal), 1 (OROM), and 2 (Secure). Default=0

DbEdition: Edition of the database. If 'Express Edition', monitor dbSize.

DbSize: Size of the database in MB. Monitor this value if the edition is 'Express Edition' and this value is close to reaching the 4000 MB maximum. Default=0

MapiVersionNumber: Internal version number of the system. It is a number that is increaed by 100 for each major and minor release. Point releases are the numbers between each 100. Value is 0 when the system does not support MapiVersionNumber. Default=0

PvsVersion[]: If successful, the PvsVersion object(s) are returned.

EXAMPLE 1: Get PvsVersion

Get the PvsVersion for the PVS SoapServer connected to.

```
Get-PvsVersion
```

EXAMPLE 2: Get the PvsVersion MapiVersion

Get the PvsVersion for the PVS SoapServer connected to.

```
Get-PvsVersion -Fields MapiVersion
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Get-PvsVirtualHostingPool

Get the fields for a Virtual Hosting Pool or all Virtual Hosting Pools in a Site or the whole Farm. All Virtual Hosting Pools are returned if no parameters are passed.

One of these optional

Guid[] Guid or VirtualHostingPoolId: GUID of the Virtual Hosting Pool to Get.

Guid[] SiteId: GUID of the Site to Get all Virtual Hosting Pools for.

string[] SiteName: Name of the Site to Get all Virtual Hosting Pools for.

or this optional & resolution

string[] Name or VirtualHostingPoolName: Name of the Virtual Hosting Pool to Get.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to Get all Virtual Hosting Pools for.

string[] SiteName: Name of the Site to Get all Virtual Hosting Pools for.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

VirtualHostingPoolId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or VirtualHostingPoolId: Read-only GUID that uniquely identifies this Virtual Hosting Pool.

Name or VirtualHostingPoolName: Name of the Virtual Hosting Pool. It is unique within the Site. Max Length=50

SiteId: GUID of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteName.

SiteName: Name of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteId.

Type: Type of the Virtual Hosting Pool. 0 = Citrix XenServer, 1 = Microsoft SCVMM/Hyper-V, 2 = VMWare vSphere/ESX. Min=0, Max=3, Default=0

Description: User description. Default="" Max Length=250

Server: Name or IP of the Host Server. Max Length=255

Port: Port of the Host Server. Min=80, Max=65534, Default=80

Datacenter: Datacenter of the Virtual Hosting Pool. Default="" Max Length=250

UpdateLimit: Number of updates at the same time. Min=2, Max=1000, Default=1000

UpdateTimeout: Timeout for updates. Min=2, Max=240, Default=60

ShutdownTimeout: Timeout for shutdown. Min=2, Max=30, Default=10

UserName: Name to use when logging into the Server.

Password: Password to use when logging into the Server.

XdHostingUnitUuid: UUID of XenDesktop Hosting Unit Default=00000000-0000-0000-0000-000000000000

PrepopulateEnabled: Enable prepopulate when set to true Default=false

XsPvsSiteUuid: UUID of XenServer PVS_site Default=00000000-0000-0000-0000-000000000000

PlatformVersion: Hypervisor Host Version Default="" Max Length=250

XdHcHypervisorConnectionName: Hypervisor Connection Name for HCL Connection Details object Default="" Max Length=250

XdHcHypervisorConnectionUid: Hypervisor Connection Uid for HCL Connection Details object Default="" Max Length=250

XdHcRevision: Revision for HCL Connection Details object Default="" Max Length=250

XdHcCustomProperties: Custom Properties for HCL Connection Details object Default="" Max Length=250

XdHcSslThumbprints: Ssl Thumbprints for HCL Connection Details object Default="" Max Length=250

PvsVirtualHostingPool[]: If successful, the PvsVirtualHostingPool object(s) are returned.

EXAMPLE 1: Get PvsVirtualHostingPool for Farm

Get all PvsVirtualHostingPool for the Farm.

```
Get-PvsVirtualHostingPool
```

EXAMPLE 2: Get PvsVirtualHostingPool for Site

Get all PvsVirtualHostingPool for the Site named theSite.

```
Get-PvsVirtualHostingPool -SiteName theSite
```

EXAMPLE 3: Get PvsVirtualHostingPool

Get PvsVirtualHostingPool for VirtualHostingPool named theVirtualHostingPool in Site named theSite.

```
Get-PvsVirtualHostingPool -Name theVirtualHostingPool -SiteName theSite
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Get-PvsXDSite

Get the fields for a XenDesktop Site or all XenDesktop Sites. All XenDisktop Sites are returned if no parameters are passed.

This optional

Guid[] Guid or XdSiteId: GUID of the XenDesktop Site to Get.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

XdSiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Guid or XdSiteId: GUID of the XenDesktop Site.

ConfigServices: XenDesktop Server addresses. Max Length=2000

PvsXDSite[]: If successful, the PvsXDSite object(s) are returned.

EXAMPLE 1: Get PvsXDSite for Farm

Get all PvsXDSite for the Farm.

```
Get-PvsXDSite
```

EXAMPLE 2: Get PvsXDSite

Get the PvsXDSite for the XDSite with Guid 45687f34-c9ec-4852-9d55-337a1af41405.

```
Get-PvsXDSite -Guid "45687f34-c9ec-4852-9d55-337a1af41405"
```

Grant-PvsAuthGroup

Assign an AuthGroup to have Farm, Site or Collection Authorization. If no Site or Collection is specified, the AuthGroup is given Farm Authorization.

One of these required

Guid[] Guid or AuthGroupId: GUID of the AuthGroup to assign Authorization for.

string[] Name or AuthGroupName: Name of the AuthGroup to assign Authorization for.

One of these optional

Guid[] SiteId: GUID of the Site to assign Authorization to for the AuthGroup.

string[] SiteName: Name of the Site to assign Authorization to for the AuthGroup.

Guid[] CollectionId: GUID of the Collection to assign Authorization to for the AuthGroup.

or this optional & resolution

string[] CollectionName: Name of the Collection to assign Authorization to for the AuthGroup.

Optional

uint Role: Authorization Role for the Collection. 300 or 400 can be used. Role can only be used with CollectionId or CollectionName. 300 is Collection Administrator, and 400 is Collection Operator. Default=400

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to assign Authorization to for the AuthGroup.

string[] SiteName: Name of the Site to assign Authorization to for the AuthGroup.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

AuthGroupId, SiteId or CollectionId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Grant-PvsAuthGroup for PvsAuthGroup to PvsFarm

```
Grant-PvsAuthGroup -Name theAuthGroup
```

EXAMPLE 2: Grant-PvsAuthGroup for PvsAuthGroup Using Pipe

The Get-PvsAuthGroup output is piped to the Grant-PvsAuthGroup.

```
Get-PvsAuthGroup -Name theAuthGroup -Fields Guid | Grant-PvsAuthGroup
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Grant-PvsAuthGroup for PvsAuthGroup to PvsSite

```
Grant-PvsAuthGroup -Name theAuthGroup -SiteName theSite
```

EXAMPLE 4: Grant-PvsAuthGroup for PvsAuthGroup Using Pipe

The Get-PvsAuthGroup output is piped to the Grant-PvsAuthGroup.

```
Get-PvsAuthGroup -Name theAuthGroup -Fields Guid | Grant-PvsAuthGroup
-SiteName theSite
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 5: Grant-PvsAuthGroup for PvsSite Using Pipe

The Get-PvsSite output is piped to the Grant-PvsAuthGroup.

```
Get-PvsSite -Name theSite -Fields Guid | Grant-PvsAuthGroup -Name
theAuthGroup
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 6: Grant-PvsAuthGroup for PvsAuthGroup to PvsCollection

```
Grant-PvsAuthGroup -Name theAuthGroup -CollectionName theCollection -
SiteName theSite
```

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 7: Grant-PvsAuthGroup for PvsAuthGroup Using Pipe

```
The Get-PvsAuthGroup output is piped to the Grant-PvsAuthGroup.

Get-PvsAuthGroup -Name theAuthGroup -Fields Guid | Grant-PvsAuthGroup
                 -CollectionName theCollection -SiteName theSite

The -Fields parameter with only the needed fields specified makes the
                 Get work faster because only those fields are
                 retrieved.

CollectionId can be used instead of CollectionName so that the
                 SiteName or SiteId is not also needed.
```

EXAMPLE 8: Grant-PvsAuthGroup for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Grant-PvsAuthGroup.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                 Grant-PvsAuthGroup -Name theAuthGroup

The -Fields parameter with only the needed fields specified makes the
                 Get work faster because only those fields are
                 retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                 also needed.
```

EXAMPLE 9: Grant-PvsAuthGroup for PvsAuthGroup to PvsCollection

```
Grant-PvsAuthGroup -Name theAuthGroup -CollectionName theCollection -
                 SiteName theSite -Role 300

CollectionId can be used instead of CollectionName so that the
                 SiteName or SiteId is not also needed.
```

EXAMPLE 10: Grant-PvsAuthGroup for PvsAuthGroup Using Pipe

```
The Get-PvsAuthGroup output is piped to the Grant-PvsAuthGroup.

Get-PvsAuthGroup -Name theAuthGroup -Fields Guid | Grant-PvsAuthGroup
                 -CollectionName theCollection -SiteName theSite -Role
                 300

The -Fields parameter with only the needed fields specified makes the
                 Get work faster because only those fields are
                 retrieved.

CollectionId can be used instead of CollectionName so that the
                 SiteName or SiteId is not also needed.
```

EXAMPLE 11: Grant-PvsAuthGroup for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Grant-PvsAuthGroup.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                 Grant-PvsAuthGroup -Name theAuthGroup -Role 300

The -Fields parameter with only the needed fields specified makes the
                 Get work faster because only those fields are
                 retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                 also needed.
```

Import-PvsDevices

Import Devices from the contents of the comma or tab delimited fileName specified. Each record needs to have Device Name, Mac Address, Site Name, Collection Name, optional Description and optional Type. Description must exist for Type to be included, but it can have 0 length. Type can be 1 when it performs test of Disks, 2 when it performs maintenance on Disks, and 0 otherwise.

This required

string[] Name or FileName: Name of the file to import the Devices from. This must be a full file path name.

One of these optional

Guid[] CollectionId: GUID of the Collection to import the Devices into.

Guid[] SiteId: GUID of the Site to import the Devices into.

string[] SiteName: Name of the Site to import the Devices into.

or this optional & resolution

string[] CollectionName: Name of the Collection to import the Devices into.

Optional

SwitchParameter CopyTemplate: If -CopyTemplate is specified, the Template Device for the collection, if it exists, will be used for the property settings of the imported Devices.

SwitchParameter DoNotCreateNewSites: If -DoNotCreateNewSites is specified, new Sites found in the file will not be created.

SwitchParameter DoNotCreateNewCollections: If -DoNotCreateNewCollections is specified, new Collections found in the file will not be created.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to import the Devices into.

string[] SiteName: Name of the Site to import the Devices into.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

CollectionId or SiteId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Import-PvsDevices

```
Import-PvsDevices -Name "C:\import\theFileName"
```

EXAMPLE 2: Import-PvsDevices

```
Import-PvsDevices -CopyTemplate -DoNotCreateNewSites -
                  DoNotCreateNewCollections -Name
                  "C:\import\theFileName"
```

EXAMPLE 3: Import-PvsDevices for SiteName

```
Import-PvsDevices -SiteName theSite -Name "C:\import\theFileName"
```

EXAMPLE 4: Import-PvsDevices for SiteName with CopyTemplate and DoNotCreateNewCollections

```
Import-PvsDevices -SiteName theSite -CopyTemplate -
                  DoNotCreateNewCollections -Name
                  "C:\import\theFileName"
```

EXAMPLE 5: Import-PvsDevices for PvsSite Using Pipe

```
The Get-PvsSite output is piped to the Import-PvsDevices.

Get-PvsSite -Name theSite -Fields Guid | Import-PvsDevices -Name
                  "C:\import\theFileName"

The -Fields parameter with only the needed fields specified makes the
                  Get work faster because only those fields are
                  retrieved.
```

EXAMPLE 6: Import-PvsDevices for CollectionName

```
Import-PvsDevices -CollectionName theCollection -SiteName theSite -
                  Name "C:\import\theFileName"

CollectionId can be used instead of CollectionName so that the
                  SiteName or SiteId is not also needed.
```

EXAMPLE 7: Import-PvsDevices for CollectionName with CopyTemplate

```
Import-PvsDevices -CollectionName theCollection -SiteName theSite -
                  CopyTemplate -Name "C:\import\theFileName"

CollectionId can be used instead of CollectionName so that the
                  SiteName or SiteId is not also needed.
```

EXAMPLE 8: Import-PvsDevices for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Import-PvsDevices.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                  Import-PvsDevices -Name "C:\import\theFileName"

The -Fields parameter with only the needed fields specified makes the
                  Get work faster because only those fields are
                  retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                  also needed.
```

Import-PvsDisk

```
Import a Disk. It will add a Disk Locator for the Disk to the Site. A
  manifest file must exist in the Store. If successful, the new
  PvsDiskLocator is returned.
```

This required & resolution

string Name or DiskLocatorName: Name of the Disk Locator File. It is unique within the Store. ASCII Max Length=52

One of these optional

Guid ServerId: GUID of the single Server that this Disk Locator is assigned to. It is not used with ServerName. Default=00000000-0000-0000-0000-000000000000

string ServerName: Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=""

Optional

string Description: User description. Default="" Max Length=250

string MenuText: Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default="" ASCII Max Length=64

SwitchParameter Enabled: True when this disk can be booted, false otherwise. Default=true

SwitchParameter RebalanceEnabled: True when this Server can automatically rebalance Devices, false otherwise. Default=false

uint RebalanceTriggerPercent: Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25

uint SubnetAffinity: Qualifier for subnet affinity when assigning a Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2, Default=0

SwitchParameter VHDX: If -VHDX is specified, VHDX will be used for the format of the image. VHDX has a Block size of 32 MB. VHD is the default.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

One of these resolutions when needed

Guid StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsDiskLocator: If successful, the new PvsDiskLocator object is returned.

EXAMPLE 1: Import-PvsDisk for VHDX to PvsDiskLocator

```
This example imports a VHDX Disk and uses the default settings for all
                other optional parameters.

$thePvsDiskLocator = Import-PvsDisk -Name theDiskLocator -SiteName
                theSite -StoreName theStore -VHDX

if ($thePvsDiskLocator -eq $null)      # check that the PvsDiskLocator
                was returned

{

     $thePvsDiskLocator.Name            # display the name

}
```

EXAMPLE 2: Import-PvsDisk for VHD to PvsDiskLocator

```
This example imports a VHD Disk and sets all of the other optional
                parameters to non-default values.

$thePvsDiskLocator = Import-PvsDisk -Name theDiskLocator -SiteName
                theSite -StoreName theStore -ServerName theServer -
                Description "The VHD disk." -MenuText "The VHD disk."
                -Enabled -RebalanceEnabled -rebalanceTriggerPercent
                50 -SubnetAffinity 2

if ($thePvsDiskLocator -eq $null)      # check that the PvsDiskLocator
                was returned

{

     $thePvsDiskLocator.Name            # display the name

}
```

Import-PvsOemLicenses

Oem Only: Import the Oem Licenses from the contents of the fileName specified.

This required

string[] Name or FileName: Name of the file to import the Oem Licenses from. This must be a full file path name.

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Import-PvsOemLicenses

```
Import-PvsOemLicenses -Name "C:\import\theFileName"
```

Invoke-PvsActivateDeviceMAK

Proxy Activate with a Multiple Activation Key and/or apply the Confirmation ID to remote activate a Device DiskLocator pair.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Activate.

string[] Name or DeviceName: Name of the Device to Activate.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Activate.

This required

string[] MakUsedToActivate: Multiple Activation Key to Activate the Device with.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Invoke-ActivateDeviceMAK for Name

```
Invoke-ActivateDeviceMAK -Name theDevice -MakUsedToActivate "2FKWD-NYFC7-VH8G3-BK3GD-7T667"
```

EXAMPLE 2: Invoke-ActivateDeviceMAK for DeviceMac

```
Invoke-ActivateDeviceMAK -DeviceMac "00-11-22-33-44-55" -MakUsedToActivate "2FKWD-NYFC7-VH8G3-BK3GD-7T667"
```

EXAMPLE 3: Invoke-ActivateDeviceMAK for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Invoke-ActivateDeviceMAK.

```
Get-PvsDevice -Name theDevice -Fields Guid | Invoke-ActivateDeviceMAK -MakUsedToActivate "2FKWD-NYFC7-VH8G3-BK3GD-7T667"
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Invoke-PvsMarkDown

Mark Down a Device, Collection, View, Server or Site.

One of these required

Guid[] DeviceId: GUID of the Device to Mark Down.

string[] DeviceName: Name of the Device to Mark Down.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Mark Down.

Guid[] CollectionId: GUID of the Collection to Mark Down all Devices.

Guid[] SiteViewId: GUID of the Site View to Mark Down all Devices.

Guid[] SiteId: GUID of the Site. Can be used alone to Mark Down all Servers and Devices in the Site.

string[] SiteName: Name of the Site. Can be used alone to Mark Down all Servers and Devices in the Site.

Guid[] FarmViewId: GUID of the Farm View to Mark Down all Devices.

string[] FarmViewName: Name of the Farm View to Mark Down all Devices.

Guid[] DiskLocatorId: GUID of the DiskLocator to Mark Down all Devices.

Guid[] ServerId: GUID of the Server to Mark Down.

string[] ServerName: Name of the Server to Mark Down.

or one of these required & resolutions

string[] CollectionName: Name of the Collection to Mark Down all Devices.

string[] SiteViewName: Name of the Site View to Mark Down all Devices.

string[] DiskLocatorName: Name of the DiskLocator to Mark Down all Devices.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site. Can be used alone to Mark Down all Servers and Devices in the Site.

string[] SiteName: Name of the Site. Can be used alone to Mark Down all Servers and Devices in the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId, CollectionId, SiteViewId, SiteId, FarmViewId, DiskLocatorId or ServerId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Invoke-PvsMarkDown for SiteName

```
Invoke-PvsMarkDown -SiteName theSite
```

EXAMPLE 2: Invoke-PvsMarkDown for PvsSite Using Pipe

The Get-PvsSite output is piped to the Invoke-PvsMarkDown.

```
Get-PvsSite -Name theSite -Fields Guid | Invoke-PvsMarkDown
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 3: Invoke-PvsMarkDown for ServerName

```
Invoke-PvsMarkDown -ServerName theServer
```

EXAMPLE 4: Invoke-PvsMarkDown for PvsServer Using Pipe

```
The Get-PvsServer output is piped to the Invoke-PvsMarkDown.

Get-PvsServer -Name theServer -Fields Guid | Invoke-PvsMarkDown

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 5: Invoke-PvsMarkDown for DeviceName

```
Invoke-PvsMarkDown -DeviceName theDevice
```

EXAMPLE 6: Invoke-PvsMarkDown for PvsDevice Using Pipe

```
The Get-PvsDevice output is piped to the Invoke-PvsMarkDown.

Get-PvsDevice -Name theDevice -Fields Guid | Invoke-PvsMarkDown

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 7: Invoke-PvsMarkDown for DeviceMac

```
Invoke-PvsMarkDown -DeviceMac "00-11-22-33-44-55"
```

EXAMPLE 8: Invoke-PvsMarkDown for PvsDevice Using Pipe

```
The Get-PvsDevice output is piped to the Invoke-PvsMarkDown.

Get-PvsDevice -DeviceMac "00-11-22-33-44-55" -Fields Guid | Invoke-
                PvsMarkDown

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 9: Invoke-PvsMarkDown for FarmViewName

```
Invoke-PvsMarkDown -FarmViewName theFarmView
```

EXAMPLE 10: Invoke-PvsMarkDown for PvsFarmView Using Pipe

```
The Get-PvsFarmView output is piped to the Invoke-PvsMarkDown.

Get-PvsFarmView -Name theFarmView -Fields Guid | Invoke-PvsMarkDown

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 11: Invoke-PvsMarkDown for CollectionName

```
Invoke-PvsMarkDown -CollectionName theCollection -SiteName theSite

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 12: Invoke-PvsMarkDown for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Invoke-PvsMarkDown.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                  Invoke-PvsMarkDown

The -Fields parameter with only the needed fields specified makes the
                  Get work faster because only those fields are
                  retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                  also needed.
```

EXAMPLE 13: Invoke-PvsMarkDown for SiteViewName

```
Invoke-PvsMarkDown -SiteViewName theSiteView -SiteName theSite

SiteViewId can be used instead of SiteViewName so that the SiteName or
                  SiteId is not also needed.
```

EXAMPLE 14: Invoke-PvsMarkDown for PvsSiteView Using Pipe

```
The Get-PvsSiteView output is piped to the Invoke-PvsMarkDown.

Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
                  Invoke-PvsMarkDown

The -Fields parameter with only the needed fields specified makes the
                  Get work faster because only those fields are
                  retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                  also needed.
```

EXAMPLE 15: Invoke-PvsMarkDown for DiskLocatorName

```
Invoke-PvsMarkDown -DiskLocatorName theDiskLocator -SiteName theSite -
                  StoreName theStore

DiskLocatorId can be used instead of DiskLocatorName so that the
                  SiteName or SiteId and StoreName or StoreId are not
                  also needed.
```

EXAMPLE 16: Invoke-PvsMarkDown for PvsDiskLocator Using Pipe

```
The Get-PvsDiskLocator output is piped to the Invoke-PvsMarkDown.

Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                  theStore -Fields Guid | Invoke-PvsMarkDown

The -Fields parameter with only the needed fields specified makes the
                  Get work faster because only those fields are
                  retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and
                  StoreName or StoreId are not also needed.
```

Invoke-PvsPromoteDiskVersion

```
Commit the changes made in the current Maintenance or a Test version.
  Promotes the Maintenance version or a Test version to a Test or new
  Production version.

   This required
```

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator File to Promote the Disk Version of.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator File to Promote the Disk Version of.

Optional

DateTime ScheduledDate: Date/Time the new disk version will become available. Uses only the date, hour and minute.

uint TestVersion: Specifies the Test version number that should be Promoted to Production.

SwitchParameter Test: If -Test is specified, set the mode of the Maintenance version to Test.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Invoke-PvsPromoteDiskVersion to Production

```
Invoke-PvsPromoteDiskVersion -Name theDiskLocator -SiteName theSite -
StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 2: Invoke-PvsPromoteDiskVersion to Production Using Pipe

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Fields Guid | Invoke-PvsPromoteDiskVersion
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 3: Invoke-PvsPromoteDiskVersion to Test

```
Invoke-PvsPromoteDiskVersion -Name theDiskLocator -SiteName theSite -
                StoreName theStore -Test
```

```
Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 4: Invoke-PvsPromoteDiskVersion to Test Using Pipe

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Invoke-PvsPromoteDiskVersion
                -Test
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

```
Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 5: Invoke-PvsPromoteDiskVersion Test to Production

```
Invoke-PvsPromoteDiskVersion -Name theDiskLocator -SiteName theSite -
                StoreName theStore -TestVersion 4
```

```
Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 6: Invoke-PvsPromoteDiskVersion Test to Production Using Pipe

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Invoke-PvsPromoteDiskVersion
                -TestVersion 4
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

```
Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 7: Invoke-PvsPromoteDiskVersion to Production, for Future

```
Invoke-PvsPromoteDiskVersion -Name theDiskLocator -SiteName theSite -
                StoreName theStore -ScheduledDate "2016/01/01"
```

```
Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 8: Invoke-PvsPromoteDiskVersion to Production, for Future Using Pipe

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Invoke-PvsPromoteDiskVersion
                -ScheduledDate "2016/01/01"
```

```
The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

```
Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

Invoke-PvsRebalanceDevices

Rebalance Devices for a Server. When successful, returns the number of Devices affected.

One of these required

Guid Guid or ServerId: GUID of the Server to Rebalance Devices on, ServerName.

string Name or ServerName: Name of the Server to Rebalance Devices on, ServerId.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

UInt32: If successful, the numeric value is returned

EXAMPLE 1: Invoke-PvsRebalanceDevices for Name

```
Invoke-PvsRebalanceDevices -Name theServer
```

Invoke-PvsRevertDiskVersion

Set the existing highest version disk to Maintenance or Test mode. A specified version can be reverted to Test mode if there are no Production versions higher than it. If the mode is Test, it can be set to Maintenance.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator File to Revert.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator File to Revert.

Optional

uint Version: Specifies the version number that should be Reverted to Test mode.

SwitchParameter Test: If -Test is specified, when reverting the highest version, the access will be set to Test, otherwise set it to Maintenance.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

```
    Guid[] StoreId: GUID of the Store that is needed when a
                    DiskLocatorName is used.

    string[] StoreName: Name of the Store that is needed when a
                    DiskLocatorName is used.

  Instead of a parameter that matches one of the members listed

    PvsObject[] Object: PvsObjects with the members below can be used as
                    the Object parameter or from a pileline:

     DiskLocatorId

  Optional

    SwitchParameter Confirm: The impact of this operation is "medium". If
                    -Confirm is specified, the operation will be
                    confirmed. $ConfirmPreference can be set to "medium"
                    or "low" to have confirmation without the Confirm
                    parameter.
```

EXAMPLE 1: Invoke-PvsRevertDiskVersion to Maintenance

```
    Invoke-PvsRevertDiskVersion -Name theDiskLocator -SiteName theSite -
                    StoreName theStore

    Guid can be used instead of Name so that the SiteName or SiteId and
                    StoreName or StoreId are not also needed.
```

EXAMPLE 2: Invoke-PvsRevertDiskVersion to Maintenance Using Pipe

```
    Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                    theStore -Fields Guid | Invoke-PvsRevertDiskVersion

    The -Fields parameter with only the needed fields specified makes the
                    Get work faster because only those fields are
                    retrieved.

    Guid can be used instead of Name so that the SiteName or SiteId and
                    StoreName or StoreId are not also needed.
```

EXAMPLE 3: Invoke-PvsRevertDiskVersion to Test

```
    Invoke-PvsRevertDiskVersion -Name theDiskLocator -SiteName theSite -
                    StoreName theStore -Test

    Guid can be used instead of Name so that the SiteName or SiteId and
                    StoreName or StoreId are not also needed.
```

EXAMPLE 4: Invoke-PvsRevertDiskVersion to Test Using Pipe

```
    Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                    theStore -Fields Guid | Invoke-PvsRevertDiskVersion -
                    Test

    The -Fields parameter with only the needed fields specified makes the
                    Get work faster because only those fields are
                    retrieved.

    Guid can be used instead of Name so that the SiteName or SiteId and
                    StoreName or StoreId are not also needed.
```

EXAMPLE 5: Invoke-PvsRevertDiskVersion Production to Test

```
    Invoke-PvsRevertDiskVersion -Name theDiskLocator -SiteName theSite -
                    StoreName theStore -Version 4
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 6: Invoke-PvsRevertDiskVersion Production to Test Using Pipe

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Fields Guid | Invoke-PvsRevertDiskVersion -
Version 4
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Merge-PvsDisk

Merge the Disk.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator File to Merge.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator File to Merge.

Optional

uint Access: Access to set the version to when merge is finished. Default value is found in the Farm MergeMode setting. Values are: 0 (Production), 1 (Test) and 2 (Maintenance), Min=0, Max=2

SwitchParameter NewBase: If -NewBase is specified, create a new base from last base plus all updates from that base. The default is to merge all updates from the last base by default.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium"

or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Merge-PvsDisk for Name

```
Merge-PvsDisk -Name theDiskLocator -SiteName theSite -StoreName
    theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 2: Merge-PvsDisk for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Merge-PvsDisk.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
    theStore -Fields Guid | Merge-PvsDisk
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 3: Merge-PvsDisk for Name with Access and NewBase

```
Merge-PvsDisk -Name theDiskLocator -SiteName theSite -StoreName
    theStore -Access 2 -NewBase
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 4: Merge-PvsDisk for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Merge-PvsDisk.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
    theStore -Fields Guid | Merge-PvsDisk -Access 2 -
    NewBase
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Mount-PvsDisk

Mount a disk. If successful, the drive letter or an empty string is returned. An empty string can be returned if a drive letter was not assigned by the operating system before the maxDiskLetterWaitSeconds is used up.

This required

Guid Guid or DiskLocatorId: GUID of the Disk Locator to Mount the Disk.

or this required & resolution

string Name or DiskLocatorName: Name of the Disk Locator to Mount the Disk.

```
One of these optional

 Guid ServerId: Specific Server GUID to use to Mount the Disk.

 string ServerName: Specific Server Name to use to Mount the Disk.

Optional

 uint MaxDiskLetterWaitSeconds: Once mapping a disk is successful, this
                 is the maximum amount of seconds spent waiting for
                 the operating system to return a drive letter. If the
                 operating system does not return a drive letter
                 before the maximum wait time, then an empty string is
                 returned. Default=30

One of these resolutions when needed

 Guid SiteId: GUID of the Site.

 string SiteName: Name of the Site.

One of these resolutions when needed

 Guid StoreId: GUID of the Store that is needed when a DiskLocatorName
                 is used.

 string StoreName: Name of the Store that is needed when a
                 DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

 PvsObject[] Object: PvsObjects with the members below can be used as
                 the Object parameter or from a pileline:

  DiskLocatorId or ServerId

Optional

 SwitchParameter Confirm: The impact of this operation is "low". If -
                 Confirm is specified, the operation will be
                 confirmed. $ConfirmPreference can be set to "low" to
                 have confirmation without the Confirm parameter.

String: If successful, the String value is returned.
```

EXAMPLE 1: Mount-PvsDisk

```
 $theDriveLetter = Start-PvsMapDisk -Name theDiskLocator -SiteName
                 theSite -StoreName theStore

 if ($theDriveLetter -ne $null -and $theDriveLetter.length -gt 0)

 {

      $theDriveLetter    # display the drive letter

 }

 Guid can be used instead of Name so that the SiteName or SiteId and
                 StoreName or StoreId are not also needed.
```

EXAMPLE 2: Mount-PvsDisk for PvsServer with MaxDiskLetterWaitSeconds

```
 $theDriveLetter = Start-PvsMapDisk -Name theDiskLocator -SiteName
                 theSite -StoreName theStore -ServerName theServer -
                 MaxDiskLetterWaitSeconds 60

 if ($theDriveLetter -ne $null -and $theDriveLetter.length -gt 0)
```

```
{
     $theDriveLetter    # display the drive letter
}
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Move-PvsDeviceToCollection

Move a Device to a Collection. Personal vDisk Devices cannot be moved to a Collection in another Site.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Move.

string[] Name or DeviceName: Name of the Device to Move.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Move.

This required

Guid[] CollectionId: GUID of the Collection to Move a Device to. The Device is moved from whatever Collection it is currently in, to the Collection specified.

or this required & resolution

string[] CollectionName: Name of the Collection to Move a Device to. The Device is moved from whatever Collection it is currently in, to the Collection specified.

Optional

SwitchParameter CopyTemplate: If -CopyTemplate is specified, the Template Device for the Collection, if it exists, will be used for the property settings of the moved Device.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId or CollectionId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Move-PvsDeviceToCollection for PvsDevice to PvsCollection

```
Move-PvsDeviceToCollection -Name theDevice -CollectionName
theCollection -SiteName theSite
```

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Move-PvsDeviceToCollection for PvsDevice Using Pipe

```
The Get-PvsDevice output is piped to the Move-PvsDeviceToCollection.

Get-PvsDevice -Name theDevice -Fields Guid | Move-
                PvsDeviceToCollection -CollectionName
                thetheCollection -SiteName theSite

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 3: Move-PvsDeviceToCollection for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Move-
                PvsDeviceToCollection.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                Move-PvsDeviceToCollection -Name theDevice

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

Move-PvsServerToSite

```
Move a Server to a Site.

    One of these required

     Guid[] Guid or ServerId: GUID of the Server to Assign.

     string[] Name or ServerName: Name of the Server to Assign.

    One of these required

     Guid[] SiteId: GUID of the Site to Assign a Server.

     string[] SiteName: Name of the Site to Assign a Server.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      ServerId or SiteId

    Optional

     SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "low" to
                have confirmation without the Confirm parameter.
```

EXAMPLE 1: Move-PvsServerToSite for PvsServer to PvsSite

```
Move-PvsServerToSite -Name theServer -SiteName theSite
```

EXAMPLE 2: Move-PvsServerToSite for PvsServer Using Pipe

```
The Get-PvsServer output is piped to the Move-PvsServerToSite.
```

```
Get-PvsServer -Name theServer -Fields Guid | Move-PvsServerToSite -
    SiteName theSite

The -Fields parameter with only the needed fields specified makes the
    Get work faster because only those fields are
    retrieved.
```

EXAMPLE 3: Move-PvsServerToSite for PvsSite Using Pipe

```
The Get-PvsSite output is piped to the Move-PvsServerToSite.

Get-PvsSite -Name theSite -Fields Guid | Move-PvsServerToSite -Name
    theServer

The -Fields parameter with only the needed fields specified makes the
    Get work faster because only those fields are
    retrieved.
```

New-PvsAuthGroup

```
Create a new authorization AuthGroup for an Active Directory or Windows
  Group.

  All parameters that do not have a Default are required, unless only a
    few of a group are required.

  string Name or AuthGroupName: Name of the Active Directory or Windows
      Group. Max Length=450

  string Description: User description. Default="" Max Length=250

  SwitchParameter Confirm: The impact of this operation is "low". If -
      Confirm is specified, the operation will be
      confirmed. $ConfirmPreference can be set to "low" to
      have confirmation without the Confirm parameter.

PvsAuthGroup: If successful, the new PvsAuthGroup object is returned.
```

EXAMPLE 1: Create PvsAuthGroup with Minimum Fields

```
Creates a PvsAuthGroup for the "AD\PVSFarmAdminGroup" Active Directory
    security group.

New-PvsAuthGroup -Name "AD\PVSFarmAdminGroup"
```

EXAMPLE 2: Create PvsAuthGroup with All Fields

```
Creates a PvsAuthGroup for the "AD\PVSFarmAdminGroup" Active Directory
    security group with "Farm AuthGroup" as the
    description.

New-PvsAuthGroup -Name "AD\PVSFarmAdminGroup" -Description "Farm
    AuthGroup"
```

EXAMPLE 3: Create PvsAuthGroup and Assign to Site

```
Creates a PvsAuthGroup for the "AD\PVSSiteAdminGroup" Active Directory
    security group and Assigns it to theSite.

New-PvsAuthGroup -Name "AD\PVSSiteAdminGroup" | Grant-PvsAuthGroup -
    SiteName theSite
```

New-PvsCeipData

Create a new entry for CeipData table.

All parameters that do not have a Default are required, unless only a few of a group are required.

uint Enabled: 1 if CEIP is enabled, otherwise 0. Min=0, Max=1

DateTime NextUpload: Date and time next CEIP upload is due if enabled is 1. Default=Empty

uint InProgress: 1 if an upload is currently in progress, otherwise 0. Default=0

Guid ServerId: ID of server that is currently uploading, null if InProgress is 0. Default=00000000-0000-0000-0000-000000000000

uint OneTimeUpload: 1 to perform a one time upload. Default=0

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsCeipData: If successful, the new PvsCeipData object is returned.

EXAMPLE 1: Create PvsCeipData with Minimum Fields

Creates a PvsCeipData with enabled as false.

```
New-PvsCeipData -Enabled 0
```

EXAMPLE 2: Create PvsCeipData with All Fields

Creates a PvsCeipData with enabled as true and with all fields set different than defaults.

```
New-PvsCeipData -Enabled 1 -NextUpload "2016-01-14 15:52:00"
```

New-PvsCisData

Create a new entry for CisData table.

All parameters that do not have a Default are required, unless only a few of a group are required.

string UserName: Username used to obtain the token Max Length=255

string Path: Path where the last problem report bundle was saved Default="" Max Length=255

string Password: Password of the user required to obtain the token. This is required only by Set and Add

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsCisData: If successful, the new PvsCisData object is returned.

EXAMPLE 1: Create PvsCisData with Minimum Fields

```
Creates a PvsCisData with userName as userName and with all fields set
                different than defaults.

New-PvsCisData -UserName userName
```

New-PvsCollection

```
Create a new Collection for a Site.

    All parameters that do not have a Default are required, unless only a
      few of a group are required.

    string Name or CollectionName: Name of the Collection. It is unique
                within the Site. Max Length=50

    Guid SiteId: GUID of the Site that this Collection is a member of. It
                is not used with SiteName.

    string SiteName: Name of the Site that this Collection is a member of.
                It is not used with SiteId.

    string Description: User description. Default="" Max Length=250

    uint LastAutoAddDeviceNumber: The Device Number of the last Auto Added
                Device. Default=0

    SwitchParameter Disabled: If -Disabled is specified, the Devices in
                the Collection can not be booted. By default the
                Devices can be booted.

    string AutoAddPrefix: The string put before the Device Number for Auto
                Add. Default="" ASCII computer name characters no end
                digit Max Length=12

    string AutoAddSuffix: The string put after the Device Number for Auto
                Add. Default="" ASCII computer name characters no
                begin digit Max Length=12

    SwitchParameter NoAutoAddZeroFill: If -NoAutoAddZeroFill is specified,
                zeros will not be placed before the Device Number up
                to the AutoAddNumberLength for Auto Add.

    uint AutoAddNumberLength: The maximum length of the Device Number for
                Auto Add. This length plus the AutoAddPrefix length
                plus the AutoAddSuffix length must be less than 16.
                Required that
                ((lenautoAddPrefix+lenautoAddSuffix)+AutoAddNumberLen
                gth)<=15. Min=3, Max=9, Default=4

    SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "low" to
                have confirmation without the Confirm parameter.

    PvsCollection: If successful, the new PvsCollection object is returned.
```

EXAMPLE 1: Create PvsCollection with Minimum Fields

```
Creates a PvsCollection named theCollection in theSite.

New-PvsCollection -Name theCollection -SiteName theSite
```

EXAMPLE 2: Create PvsCollection with All Fields

```
Creates a PvsCollection named theCollection in theSite with "XenServer
                Collection" as the description that is Disabled and
                has all AutoAdd settings different than defaults.

New-PvsCollection -Name theCollection -SiteName theSite -Description
                "XenServer Collection" -Disabled -
                LastAutoAddDeviceNumber 100 -AutoAddPrefix A -
                AutoAddSuffix A -NoAutoAddZeroFill -
                AutoAddNumberLength 3
```

EXAMPLE 3: Create PvsCollection and Assign AuthGroup to it

```
Creates a PvsCollection named theCollection in theSite and Assigns
                AuthGroup "AD\PVSCollectionAdminGroup" to it.

New-PvsCollection -Name theCollection -SiteName theSite | Grant-
                PvsAuthGroup -AuthGroupName
                "AD\PVSCollectionAdminGroup"
```

New-PvsDevice

```
Add a new Device to a Collection.

    All parameters that do not have a Default are required, unless only a
      few of a group are required.

    string Name or DeviceName: Computer name with no spaces. ASCII
                computer name characters Max Length=15

    Guid CollectionId: GUID of the Collection this Device is to be a
                member of. It is not used with CollectionName.

    string CollectionName: Name of the Collection this Device is to be a
                member of. SiteName or SiteId must also be used.

    Guid SiteId: GUID of the Site the CollectionName is to be a member of.
                This or SiteName is used with CollectionName.

    string SiteName: Name of the Site the CollectionName is to be a member
                of. This or SiteId is used with CollectionName.

    string Description: User description. Default="" Max Length=250

    PvsPhysicalAddress DeviceMac: Ethernet address can have the form XX-
                XX-XX-XX-XX-XX. Uniquely identifies the Device.

    uint BootFrom: Device to boot from. Choices are 1 for vDisk, 2 for
                Hard Disk, and 3 for Floppy. Min=1, Max=3, Default=1

    string ClassName: Used by Automatic Update feature to match new
                versions of Disks to a Device. Default="" Max
                Length=41

    uint Port: UDP port to use with Stream Service. Min=1025, Max=65534,
                Default=6901

    SwitchParameter Disabled: If -Disabled is specified, the Device can
                not be booted. By default the Device can be booted.

    SwitchParameter LocalDiskEnabled: If -LocalDiskEnabled is specified,
                there will be a local disk menu choice for the
                Device.
```

uint Authentication: Device log in authentication. Choices are 0 for none, 1 for User Name/Password, and 2 for Extern. Min=0, Max=2, Default=0

string User: Name of user to authenticate before the boot process continues. Default="" ASCII Max Length=20

string Password: Password of user to authenticate before the boot process continues. Default="" ASCII Max Length=100

SwitchParameter CopyTemplate: If -CopyTemplate is specified, the Template Device for the collection, if it exists, will be used for the property settings of the new Device.

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

string AdPassword: The Active Directory machine account password. Default="" ASCII Max Length=256

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

string DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

uint Type: 1 when it performs test of Disks, 2 when it performs maintenance on Disks, 0 otherwise. Min=0, Max=2, Default=0

uint LocalWriteCacheDiskSize: The size in GB to format the Device cache file disk. If the value is 0, then the disk is not formatted. Min=0, Max=2048, Default=0

Guid VirtualHostingPoolId: GUID that uniquely identifies the Virtual Hosting Pool for a VM. This is needed when Adding a VM device. Default=00000000-0000-0000-0000-000000000000

string HypVmId: Hypervisor VM ID for HCL Default="" Max Length=250

```
        SwitchParameter BdmBoot: If -BdmBoot is 0, use PXE, 1 use BDM. PXE
                        boot is used by default.

        uint BdmType: Use PXE boot when set to 0, BDM (Bios) boot when set to
                        1 and BDM (Uefi) boot when set to 2.  Default=0

        uint BdmFormat: 1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0

        DateTime BdmUpdated: Timestamp of the last BDM boot disk update.
                        Default=Empty

        DateTime BdmCreated: Timstamp when BDM device was
                        created  Default=Empty

        Guid XsPvsProxyUuid: UUID of XenServer PVS_proxy Default=00000000-
                        0000-0000-0000-000000000000

        string EnableXsProxy: Enable XenServerProxy when set to 1 Default=""

        SwitchParameter Confirm: The impact of this operation is "low". If -
                        Confirm is specified, the operation will be
                        confirmed. $ConfirmPreference can be set to "low" to
                        have confirmation without the Confirm parameter.

    PvsDevice: If successful, the new PvsDevice object is returned.
```

EXAMPLE 1: Create PvsDevice with Minimum Fields

```
        Creates a PvsDevice named theDevice in theCollection of theSite.

        New-PvsDevice -Name theDevice -DeviceMac "00-11-22-33-44-55" -SiteName
                        theSite -CollectionName theCollection

        CollectionId can be used instead of CollectionName so that the
                        SiteName or SiteId is not also needed.
```

EXAMPLE 2: Create PvsDevice with All Fields

```
        Creates a PvsDevice named theDevice in theCollection of theSite with
                        all fields set different than defaults.

        New-PvsDevice -Name theDevice -DeviceMac "00-11-22-33-44-55" -SiteName
                        theSite -CollectionName theCollection -Description
                        "XenServer Device" -BootFrom 2 -ClassName C -Port
                        6000 -Disabled -LocalDiskEnabled -Authentication 1 -
                        User U -Password P -CopyTemplate -LogLevel 3 -Type 2
                        -LocalWriteCacheDiskSize 100 -BdmBoot -
                        VirtualHostingPoolId "15e0544e-4cf9-449e-b47f-
                        8d836b16026f"

        Do not set AdTimestamp, AdSignature, DomainName, DomainObjectSID,
                        DomainControllerName and DomainTimeCreated. They are
                        only set internally by PVS.

        CollectionId can be used instead of CollectionName so that the
                        SiteName or SiteId is not also needed.
```

EXAMPLE 3: Create PvsDevice and Boot it

```
        Creates a PvsDevice named theDevice in theCollection of theSite and
                        Boots it.

        New-PvsDevice -Name theDevice -DeviceMac "00-11-22-33-44-55" -SiteName
                        theSite -CollectionName theCollection | Start-PvsBoot
```

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

New-PvsDeviceWithPersonalvDisk

Add a new Device with Personal vDisk to a collection.

All parameters that do not have a Default are required, unless only a few of a group are required.

string Name or DeviceName: Computer name with no spaces. ASCII computer name characters Max Length=15

Guid CollectionId: GUID of the Collection this Device with Personal vDisk is to be a member of. It is not used with CollectionName.

string CollectionName: Name of the Collection this Device with Personal vDisk is to be a member of. SiteName or SiteId must also be used.

Guid DiskLocatorId: GUID of the Disk Locator to update with this Device.

Guid SiteId: GUID of the Site the CollectionName is to be a member of. This or SiteName is used with CollectionName.

string SiteName: Name of the Site the CollectionName is to be a member of. This or SiteId is used with CollectionName.

string Description: User description. Default="" Max Length=250

PvsPhysicalAddress DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device with Personal vDisk.

uint Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

string DomainName: Fully qualified name of the domain that the Device with Personal vDisk belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device with Personal vDisk's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is

only set internally by PVS. Default="" Max Length=4000

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

uint Type: 3 when it has a Personal vDisk, 4 when it has a Personal vDisk and performs tests. Min=3, Max=4, Default=3

string PvdDriveLetter: Personal vDisk Drive letter. Range is F to Z. Default="" Max Length=1

uint LocalWriteCacheDiskSize: The size in GB to format the Device cache file disk. If the value is 0, then the disk is not formatted. Min=0, Max=2048, Default=0

SwitchParameter BdmBoot: If -BdmBoot is specified, user BDM instead of PXE boot. PXE boot is used by default.

Guid XdSiteId: GUID of the XenDesktop Site. Default=00000000-0000-0000-0000-000000000000

uint XdCatalogId: Integer identifier of the XenDesktop Catalog. Default=""

Guid VirtualHostingPoolId: GUID that uniquely identifies the Virtual Hosting Pool for a VM. This is needed when Adding a VM device. Default=00000000-0000-0000-0000-000000000000

string EnableXsProxy: Enable XenServerProxy when set to 1 Default=""

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsDeviceWithPersonalvDisk: If successful, the new PvsDeviceWithPersonalvDisk object is returned.

EXAMPLE 1: Create PvsDeviceWithPersonalvDisk with Minimum Fields

Creates a PvsDeviceWithPersonalvDisk with Personal vDisk named theDevice in theCollection of theSite.

New-PvsDeviceWithPersonalvDisk -Name theDevice -DeviceMac "00-11-22-33-44-55" -SiteName theSite -CollectionName theCollection

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Create PvsDeviceWithPersonalvDisk with All Fields

Creates a PvsDeviceWithPersonalvDisk named theDevice in theCollection of theSite with all fields set different than defaults.

New-PvsDeviceWithPersonalvDisk -Name theDevice -DeviceMac "00-11-22-33-44-55" -SiteName theSite -CollectionName theCollection -Description "XenServer Device" -ClassName C -Port 6000 -LogLevel 3 -Type 4 -PvdDriveLetter G -LocalWriteCacheDiskSize 100 -

BdmBoot XdSiteId "bd712b52-a262-4aa2-9c36-1602efe07f57" -XdCatalogId 5 -VirtualHostingPoolId "15e0544e-4cf9-449e-b47f-8d836b16026f"

Do not set AdTimestamp, AdSignature, DomainName, DomainObjectSID, DomainControllerName and DomainTimeCreated. They are only set internally by PVS.

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 3: Create PvsDeviceWithPersonalvDisk and Boot it

Creates a PvsDeviceWithPersonalvDisk with Personal vDisk named theDevice in theCollection of theSite and Boots it.

New-PvsDeviceWithPersonalvDisk -Name theDevice -DeviceMac "00-11-22-33-44-55" -SiteName theSite -CollectionName theCollection | Start-PvsBoot

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

New-PvsDirectory

Create a Directory on the Server specified.

One of these required

Guid[] Guid or ServerId: GUID of the Server to create a Directory on.

string[] Name or ServerName: Name of the Server to create a Directory on.

This required

string[] Path: Path of the new Directory to create.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: New-PvsDirectory for Name

New-PvsDirectory -Name theServer -Path "C:\directory\subdirectory"

EXAMPLE 2: New-PvsDirectory for PvsServer Using Pipe

The Get-PvsServer output is piped to the New-PvsDirectory.

Get-PvsServer -Name theServer -Fields Guid | New-PvsDirectory -Path "C:\directory\subdirectory"

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

New-PvsDiskLocator

Create a new Disk Locator for a Site. The Disk file must already exist.

All parameters that do not have a Default are required, unless only a few of a group are required.

string Name or DiskLocatorName: Name of the Disk Locator File. It is unique within the Store. ASCII Max Length=52

Guid SiteId: GUID of the Site this DiskLocator is to be a member of. It is not used with SiteName.

string SiteName: Name of the Site this DiskLocator is to be a member of. It is not used with SiteId.

Guid StoreId: GUID of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreName.

string StoreName: Name of the Store that this Disk Locator is a member of. SiteName or SiteId must also be used. It is not used with StoreId.

string Description: User description. Default="" Max Length=250

string MenuText: Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default="" ASCII Max Length=64

Guid ServerId: GUID of the single Server that this Disk Locator is assigned to. It is not used with ServerName. Default=00000000-0000-0000-0000-000000000000

string ServerName: Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=""

SwitchParameter Disabled: If -Disabled is specified, the disk can not be booted. By default the disk can be booted.

SwitchParameter RebalanceEnabled: If -RebalanceEnabled is specified, this Server can automatically rebalance Devices.

uint RebalanceTriggerPercent: Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25

uint SubnetAffinity: Qualifier for subnet affinity when assigning a Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2, Default=0

string NewDiskWriteCacheType: The WriteCacheType that if a new Disk will be created, it will be set with. It is only used when a new Disk is being created. Value are: 0 (Private), (other values are standard image) 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), or 9 (Cache in Device RAM with Overflow on Hard Disk) Default=0

SwitchParameter VHDX: If -VHDX is specified, the format of the image the DiskLocator is being added for that has never been added to the Farm is VHDX. Otherwise it is assumed to be VHD.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsDiskLocator: If successful, the new PvsDiskLocator object is returned.

EXAMPLE 1: Create PvsDiskLocator with Minimum Fields

Creates a PvsDiskLocator named theDiskLocator in theSite and theStore.

```
New-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore
```

EXAMPLE 2: Create PvsDiskLocator with All Fields

Creates a PvsDiskLocator named theDiskLocator in theSite and theStore with all fields set different than defaults.

```
New-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Description "XenServer DiskLocator" -
MenuText "XenServer Disk" -ServerName theServer -
Disabled -RebalanceEnabled -RebalanceTriggerPercent -
50 -SubnetAffinity -1 -NewDiskWriteCacheType 9 -VHDX
```

-NewDiskWriteCacheType 9 (Cache in Device RAM with Overflow on Hard Disk) is an important parameter since this is the most optimal cache type.

New-PvsDiskMaintenanceVersion

Create a Maintenance version for the Disk Locator. Returns a PvsDiskVersion when successful.

This required

Guid Guid or DiskLocatorId: GUID of the Disk Locator File to Enable Disk Maintenance on.

or this required & resolution

string Name or DiskLocatorName: Name of the Disk Locator File to Enable Disk Maintenance on.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

One of these resolutions when needed

Guid StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsDiskVersion: If successful, the new PvsDiskVersion object is returned.

EXAMPLE 1: New-PvsDiskMaintenanceVersion

This example create a maintenance version for an existing DiskLocator. It returns a PvsDiskVersion when successful.

```
$thePvsDiskVersion = New-PvsDiskMaintenanceVersion -Name
theDiskLocator -SiteName theSite -StoreName theStore

if ($thePvsDiskVersion -ne $null)

{

     $thePvsDiskVersion.Name    # display the name of the version file

}
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

New-PvsDiskUpdateDevice

Add a new Device related to a Disk that can be updated.

All parameters that do not have a Default are required, unless only a few of a group are required.

string Name or DeviceName: Computer name with no spaces. ASCII computer name characters Max Length=15

Guid VirtualHostingPoolId: GUID of the Virtual Hosting Pool. It is not used with VirtualHostingPoolName.

string VirtualHostingPoolName: Name of the Virtual Hosting Pool.

Guid DiskLocatorId: GUID of the Disk Locator to update with this Device.

string Description: User description. Default="" Max Length=250

PvsPhysicalAddress DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device.

uint Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

string DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

string AdPassword: The Active Directory machine account password. Default="" Max Length=256

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsDiskUpdateDevice: If successful, the new PvsDiskUpdateDevice object is returned.

EXAMPLE 1: Create PvsDiskUpdateDevice with Minimum Fields

Creates a PvsDiskUpdateDevice named theDevice for the DiskLocatorId 2888f183-2f48-4771-b8dd-e5a44b2ee59b.

```
New-PvsDiskUpdateDevice -Name theDevice -DeviceMac "00-11-22-33-44-55"
-VirtualHostingPoolName theVirtualHostingPool -
DiskLocatorId "2888f183-2f48-4771-b8dd-e5a44b2ee59b"
```

EXAMPLE 2: Create PvsDiskUpdateDevice with All Fields

Creates a PvsDiskUpdateDevice named theDevice for the DiskLocatorId 2888f183-2f48-4771-b8dd-e5a44b2ee59b with all fields set different than defaults.

```
New-PvsDiskUpdateDevice -Name theDevice -DeviceMac "00-11-22-33-44-55"
-VirtualHostingPoolName theVirtualHostingPool -
DiskLocatorId "2888f183-2f48-4771-b8dd-e5a44b2ee59b"
-Description "DiskUpdate Device" -Port 6000 -LogLevel
3
```

Do not set AdTimestamp, AdSignature, DomainName, DomainObjectSID, DomainControllerName and DomainTimeCreated. They are only set internally by PVS.

New-PvsFarmView

Create a new View for the Farm.

All parameters that do not have a Default are required, unless only a few of a group are required.

string Name or FarmViewName: name of the Farm View. Max Length=50

string Description: User description. Default="" Max Length=250

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsFarmView: If successful, the new PvsFarmView object is returned.

EXAMPLE 1: Create PvsFarmView with Minimum Fields

Creates a PvsFarmView named theFarmView.

```
New-PvsFarmView -Name theFarmView
```

EXAMPLE 2: Create PvsFarmView with All Fields

Creates a PvsFarmView named theFarmView with "XenServer FarmView" as the description.

```
New-PvsFarmView -Name theFarmView -Description "XenServer FarmView"
```

New-PvsServer

Add a new Server to a Site.

All parameters that do not have a Default are required, unless only a few of a group are required.

string Name or ServerName: Computer name with no spaces. ASCII computer name characters Max Length=21

Guid SiteId: GUID of the Site this Server is to be a member of. It is not used with SiteName.

string SiteName: Name of the Site this Server is to be a member of. It is not used with SiteId.

string Description: User description. Default="" Max Length=250

uint AdMaxPasswordAge: Number of days before a password expires. Min=1, Max=30, Default=7

uint LicenseTimeout: Amount of seconds before a license times out. Min=15, Max=300, Default=30

uint VDiskCreatePacing: VDisk create time pacing in miliseconds. Min=0, Max=5, Default=0

uint FirstPort: Number of the first UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6910

uint LastPort: Number of the last UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6930

uint ThreadsPerPort: Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8

uint BuffersPerThread: Number of buffers per worker thread. Min=1, Max=128, Default=24

uint ServerCacheTimeout: Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8

uint IoBurstSize: Number of bytes read/writes can send in a burst of packets. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=4096, Max=61440, Default=32768

uint MaxTransmissionUnits: Ethernet maximum transmission unit size for the protocol for use for Server and Device. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=502, Max=16426, Default=1506

uint MaxBootDevicesAllowed: Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500

uint MaxBootSeconds: Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60

uint BootPauseSeconds: Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10

SwitchParameter AdMaxPasswordAgeEnabled: If -AdMaxPasswordAgeEnabled is specified, Age the password.

SwitchParameter EventLoggingEnabled: If -EventLoggingEnabled is specified, event logging is enabled.

SwitchParameter NonBlockingIoDisabled: If -NonBlockingIoDisabled is specified, do not use non-Blocking IO.

string[] Ip: One or more streaming ip addresses.

uint InitialQueryConnectionPoolSize: Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50

uint InitialTransactionConnectionPoolSize: Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50

uint MaxQueryConnectionPoolSize: Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000

uint MaxTransactionConnectionPoolSize: Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000

uint RefreshInterval: Interval, in number of seconds, the server should wait before refreshing settings. If set to 0, unused database connections are never released. Min=0, Max=32767, Default=300

uint UnusedDbConnectionTimeout: Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300

uint BusyDbConnectionRetryCount: Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2

uint BusyDbConnectionRetryInterval: Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25

```
uint LocalConcurrentIoLimit: Maximum concurrent IO transactions it
                performs for vDisks that are local. A value of 0
                disables the feature. Min=0, Max=128, Default=4

uint RemoteConcurrentIoLimit: Maximum concurrent IO transactions it
                performs for vDisks that are remote. A value of 0
                disables the feature. Min=0, Max=128, Default=4

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1
                (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug),
                and 6 (Trace). Min=0, Max=6, Default=4

uint LogFileSizeMax: Maximum size log files can reach in Megabytes.
                Min=1, Max=50, Default=5

uint LogFileBackupCopiesMax: Maximum number of log file backups.
                Min=1, Max=50, Default=4

float PowerRating: A strictly relative rating of this Server's
                capabilities when compared to other Servers in the
                Store(s) it belongs too; can be used to help tune
                load balancing. Min=0.1, Max=1000, Default=1

DateTime LastCeipUploadAttempt: Time that this server last attempted a
                CEIP upload. Default=Empty

DateTime LastBugReportAttempt: Time that this server last attempted to
                upload or generate a bug report bundle. Default=Empty

string LastBugReportStatus: Status of the last bug report on this
                server. Default="" Max Length=250

string LastBugReportResult: Status of the last bug report on this
                server. Default="" Max Length=4000

string LastBugReportSummary: Summary of the last bug report on this
                server. Default="" Max Length=250

SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "low" to
                have confirmation without the Confirm parameter.

PvsServer: If successful, the new PvsServer object is returned.
```

EXAMPLE 1: Create PvsServer with Minimum Fields

```
Creates a PvsServer named theServer in theSite.

New-PvsServer -Name theServer -SiteName theSite -Ip 192.168.0.33
                192.168.0.34
```

EXAMPLE 2: Create PvsServer with All Fields

```
Creates a PvsServer named theServer in theSite with all fields set
                different than defaults.

New-PvsServer -Name theServer -SiteName theSite -Ip 192.168.0.33
                192.168.0.34 -Description "XenServer Server" -
                AdMaxPasswordAge 10 -LicenseTimeout 60 -
                VDiskCreatePacing 5 -FirstPort 7910 -LastPort 7930 -
                ThreadsPerPort 16 -BuffersPerThread 48 -
                ServerCacheTimeout 15 -IoBurstSize 13632 -
                MaxTransmissionUnits 502 -MaxBootDevicesAllowed 1000
                -MaxBootSeconds 120 -BootPauseSeconds 20 -
```

```
AdMaxPasswordAgeEnabled -EventLoggingEnabled -
NonBlockingIoDisabled -InitialQueryConnectionPoolSize
100 -InitialTransactionConnectionPoolSize 100 -
MaxQueryConnectionPoolSize 2000 -
MaxTransactionConnectionPoolSize 2000 -
RefreshInterval 500 -UnusedDbConnectionTimeout 500 -
BusyDbConnectionRetryCount 5 -
BusyDbConnectionRetryInterval 50 -
LocalConcurrentIoLimit 5 -RemoteConcurrentIoLimit 5 -
LogLevel 3 -LogFileSizeMax 10 -LogFileBackupCopiesMax
5 -PowerRating 1.5
```

New-PvsSite

Create a new Site for the Farm.

All parameters that do not have a Default are required, unless only a few of a group are required.

string Name or SiteName: Name of the Site. Max Length=50

string Description: User description. Default="" Max Length=250

uint InventoryFilePollingInterval: The number of seconds between polls for Disk changes in the Stores. Min=1, Max=600, Default=60

SwitchParameter EnableDiskUpdate: If -EnableDiskUpdate is specified, the Disk Update will be enabled for the Site. By default Disk Update is disabled.

Guid DiskUpdateServerId: GUID of the Disk Update Server for the Site. Not used with DiskUpdateServerName. Default=00000000-0000-0000-0000-000000000000

string DiskUpdateServerName: Name of the Disk Update Server for the Site. Not used with DiskUpdateServerId. Default=""

string MakUser: User name used for MAK activation. Default="" Max Length=64

string MakPassword: User password used for MAK activation. Default="" Max Length=64

string EnableXsProxy: Enable XenServerProxy when set to 1 Default=""

Guid VirtualHostingPoolId: GUID of the VirtualHostingPool object.

string VirtualHostingPoolName: Name of the VirtualHostingPool object.

string XsPvsSiteUuid: GUID of the XenServer PVS Site.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsSite: If successful, the new PvsSite object is returned.

EXAMPLE 1: Create PvsSite with Minimum Fields

Creates a PvsSite named theSite.

```
New-PvsSite -Name theSite
```

EXAMPLE 2: Create PvsSite with All Fields

```
Creates a PvsSite named theSite with all fields set different than
                defaults.

New-PvsSite -Name theSite -Description "XenServer Site" -
                InventoryFilePollingInterval 100 -EnableDiskUpdate -
                DiskUpdateServerName theServer -MakUser theMakUser -
                MakPassword theMakPassword
```

EXAMPLE 3: Create PvsSite and Assign AuthGroup to it

```
Creates a PvsSite named theSite and Assigns AuthGroup
                "AD\PVSSiteAdminGroup" to it.

New-PvsSite -Name theSite | Grant-PvsAuthGroup -AuthGroupName
                "AD\PVSSiteAdminGroup"
```

New-PvsSiteView

```
Create a new View for a Site.

    All parameters that do not have a Default are required, unless only a
      few of a group are required.

    string Name or SiteViewName: Name of the Site View. Max Length=50

    Guid SiteId: GUID of the Site this View is to be a member of. It is
                not used with SiteName.

    string SiteName: Name of the Site this View is to be a member of. It
                is not used with SiteId.

    string Description: User description. Default="" Max Length=250

    SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "low" to
                have confirmation without the Confirm parameter.

    PvsSiteView: If successful, the new PvsSiteView object is returned.
```

EXAMPLE 1: Create PvsSiteView with Minimum Fields

```
Creates a PvsSiteView named theSiteView in theSite.

New-PvsSiteView -Name theSiteView -SiteName theSite
```

EXAMPLE 2: Create PvsSiteView with All Fields

```
Creates a PvsSiteView named theSiteView in theSite with "XenServer
                SiteView" as the description.

New-PvsSiteView -Name theSiteView -SiteName theSite -Description
                "XenServer SiteView"
```

New-PvsStore

```
Create a new Store for the Farm.

    All parameters that do not have a Default are required, unless only a
      few of a group are required.

    string Name or StoreName: Name of the Store. Max Length=50
```

Guid SiteId: GUID of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteName can be used instead. Default=00000000-0000-0000-0000-000000000000

string SiteName: Name of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteId can be used instead. Default=""

string Description: User description. Default="" Max Length=250

string Path: Default directory path that the Servers use to access this Store. Max Length=255

string[] CachePath: Default Cache path(s) that the Servers use with this Store. If none are specified the caches will be placed in the WriteCache subdirectory of the Store path. Default=None

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsStore: If successful, the new PvsStore object is returned.

EXAMPLE 1: Create PvsStore with Minimum Fields

Creates a PvsStore named theStore with network share path.

```
New-PvsStore -Name theStore -Path "\\thePath\subDirectory"
```

EXAMPLE 2: Create PvsStore with All Fields

Creates a PvsStore named theStore that only theSite administrators can change with all fields set different than defaults.

```
New-PvsStore -Name theStore -Path "c:\thePath" -SiteName theSite -Description "Local Server Path Store" -CachePath "c:\thePath\sub1" "c:\thePath\sub2"
```

New-PvsUpdateTask

Create a new Update Task for a Store.

All parameters that do not have a Default are required, unless only a few of a group are required.

string Name or UpdateTaskName: Name of the Update Task. It is unique within the Site. Max Length=50

Guid SiteId: GUID of the Site that this Update Task is a member of. It is not used with SiteName.

string SiteName: Name of the Site that this Update Task is a member of. It is not used with SiteId.

string Description: User description. Default="" Max Length=250

SwitchParameter Disabled: If -Disabled is specified, the updates will not be processed. By default the updates will be processed.

uint Hour: The hour of the day to perform the task. Min=0, Max=23, Default=0

uint Minute: The minute of the hour to perform the task. Min=0, Max=59, Default=0

uint Recurrence: The update will reoccur on this schedule. 0 = None, 1 = Daily, 2 = Every Weekday, 3 = Weekly, 4 = Monthly Date, 5 = Monthly Type. Min=0, Max=5, Default=0

uint DayMask: Days selected values. 1 = Monday, 2 = Tuesday, 4 = Wednesday, 8 = Thursday, 16 = Friday, 32 = Saturday, 64 = Sunday, 128 = Day. Default=0. This is used with Weekly and Monthly Type recurrence. Min=1, Max=255, Default=4

uint[] Date: Days of the month. Numbers from 1-31 are the only valid values. This is used with Monthly Date recurrence. Default="" Max Length=83

uint MonthlyOffset: When to happen monthly. 0 = None, 1 = First, 2 = Second, 3 = Third, 4 = Forth, 5 = Last. This is used with Monthly Type recurrence. Min=0, Max=5, Default=3

string EsdType: Esd to use. Valid values are SCCM or WSUS. If no value, a custom script is run on the client. Default="" Max Length=50

string PreUpdateScript: Script file to run before the update starts. Default="" Max Length=255

string PreVmScript: Script file to run before the VM is loaded. Default="" Max Length=255

string PostUpdateScript: Script file to run after the update finishes. Default="" Max Length=255

string PostVmScript: Script file to run after the VM is unloaded. Default="" Max Length=255

string Domain: Domain to add the Disk Update Device(s) to. If not included, the first Domain Controller found on the Server is used. Default="" Max Length=255

string OrganizationUnit: Organizational Unit to add the Disk Update Device(s) to. This parameter is optional. If it is not specified, the device is added to the built in Computers container. Child OU's should be delimited with forward slashes, e.g. "ParentOU/ChildOU". Special characters in an OU name, such as '"', '#', '+', ',', ';', '>', '=', must be escaped with a backslash. For example, an OU called "commaIn,TheMiddle" must be specified as "commaIn\,TheMiddle". The old syntax of delimiting child OU's with a comma is still supported, but deprecated. Note that in this case, the child OU comes first, e.g. "ChildOU,ParentOU". Default="" Max Length=255

uint PostUpdateApprove: Access to place the version in after the update has occurred. 0 = Production, 1 = Test, 2 = Maintenance. Min=0, Max=2, Default=0

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be

confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsUpdateTask: If successful, the new PvsUpdateTask object is returned.

EXAMPLE 1: Create PvsUpdateTask with Minimum Fields

Creates a PvsUpdateTask named theUpdateTask in theSite. Since no EsdType is set, the update.bat in the client install directory will be executed for this task.

```
New-PvsUpdateTask -Name theUpdateTask -SiteName theSite
```

EXAMPLE 2: Create PvsUpdateTask for Daily Updates

Creates a PvsUpdateTask named theUpdateTask in theSite for SCCM that occurs every day at midnight.

```
New-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Description "Every day at Midnight" -Recurrence 1 -Hour 0 -Minute 0 -EsdType SCCM
```

EXAMPLE 3: Create PvsUpdateTask for Weekday Updates

Creates a PvsUpdateTask named theUpdateTask in theSite for WSUS that occurs every weekday at midnight.

```
New-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Description "Every weekday at Midnight" -Recurrence 2 -Hour 0 -Minute 0 -EsdType WSUS
```

EXAMPLE 4: Create PvsUpdateTask for Weekly Updates

Creates a PvsUpdateTask named theUpdateTask in theSite for SCCM that occurs every week on Monday and Friday at 3 AM.

```
New-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Description "Every week on Monday and Friday at 3 AM" -Recurrence 3 -Hour 3 -Minute 0 -DayMask 17 -EsdType SCCM
```

EXAMPLE 5: Create PvsUpdateTask for Monthly Updates

Creates a PvsUpdateTask named theUpdateTask in theSite for WSUS that occurs every month on the 1st and 15th at midnight.

```
New-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Description "Every month on the 1st and 15th at Midnight" -Recurrence 4 -Hour 0 -Minute 0 -Date 1 15 -EsdType WSUS
```

EXAMPLE 6: Create PvsUpdateTask for Monthly Type Updates

Creates a PvsUpdateTask named theUpdateTask in theSite for SCCM that is disabled and occurs every month on the 2nd Tuesday and Thusday at Midnight.

```
New-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Description "Every month on the 2nd Tuesday and Thusday at Midnight" -Disabled -Recurrence 5 -Hour 0 -Minute 0 -DayMask 10 -MonthlyOffset 2 -EsdType SCCM
```

New-PvsVirtualHostingPool

Add a new Virtual Hosting Pool to a Site.

All parameters that do not have a Default are required, unless only a few of a group are required.

string Name or VirtualHostingPoolName: Name of the Virtual Hosting Pool. It is unique within the Site. Max Length=50

Guid SiteId: GUID of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteName.

string SiteName: Name of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteId.

uint Type: Type of the Virtual Hosting Pool. 0 = Citrix XenServer, 1 = Microsoft SCVMM/Hyper-V, 2 = VMWare vSphere/ESX. Min=0, Max=3, Default=0

string Description: User description. Default="" Max Length=250

string Server: Name or IP of the Host Server. Max Length=255

uint Port: Port of the Host Server. Min=80, Max=65534, Default=80

string Datacenter: Datacenter of the Virtual Hosting Pool. Default="" Max Length=250

uint UpdateLimit: Number of updates at the same time. Min=2, Max=1000, Default=1000

uint UpdateTimeout: Timeout for updates. Min=2, Max=240, Default=60

uint ShutdownTimeout: Timeout for shutdown. Min=2, Max=30, Default=10

string UserName: Name to use when logging into the Server.

string Password: Password to use when logging into the Server.

Guid XdHostingUnitUuid: UUID of XenDesktop Hosting Unit Default=00000000-0000-0000-0000-000000000000

SwitchParameter PrepopulateEnabled: Enable prepopulate when set to true Default=false

Guid XsPvsSiteUuid: UUID of XenServer PVS_site Default=00000000-0000-0000-0000-000000000000

string PlatformVersion: Hypervisor Host Version Default="" Max Length=250

string XdHcHypervisorConnectionName: Hypervisor Connection Name for HCL Connection Details object Default="" Max Length=250

string XdHcHypervisorConnectionUid: Hypervisor Connection Uid for HCL Connection Details object Default="" Max Length=250

string XdHcRevision: Revision for HCL Connection Details object Default="" Max Length=250

string XdHcCustomProperties: Custom Properties for HCL Connection Details object Default="" Max Length=250

string XdHcSslThumbprints: Ssl Thumbprints for HCL Connection Details object Default="" Max Length=250

string DisableHostXsProxy: True to disable PVS-Accelerator Default=""

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be

confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsVirtualHostingPool: If successful, the new PvsVirtualHostingPool object is returned.

EXAMPLE 1: Create PvsVirtualHostingPool with Minimum Fields

Creates a PvsVirtualHostingPool named theVirtualHostingPool in theSite.

```
New-PvsVirtualHostingPool -Name theVirtualHostingPool -SiteName
theSite -Server 192.168.0.33 -UserName theUserName -
Password thePassword
```

EXAMPLE 2: Create PvsVirtualHostingPool with All Fields

Creates a PvsVirtualHostingPool named theVirtualHostingPool in theSite with all fields set different than defaults.

```
New-PvsVirtualHostingPool -Name theVirtualHostingPool -SiteName
theSite -Server 192.168.0.33 -UserName theUserName -
Password thePassword -Type 1 -Description "A
VirtualHostingPool" -Port 180 -Datacenter Data -
UpperLimit 500 -UpdateTimeout 120 -ShutdownTimeout 20
```

Remove-PvsAuthGroup

Remove one or more AuthGroup Active Directory or Windows Group names.

One of these required

Guid[] Guid or AuthGroupId: GUID of the AuthGroup to Delete.

string[] Name or AuthGroupName: Name of the AuthGroup to Delete.

Optional

SwitchParameter Force: If -Force is specified, the AuthGroup will be Deleted even if being used, otherwise an error is returned if being used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

AuthGroupId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsAuthGroup with Name theAuthGroup

This example removes the PvsAuthGroup named theAuthGroup.

```
Remove-PvsAuthGroup -Name theAuthGroup
```

EXAMPLE 2: Remove the PvsAuthGroup with Name theAuthGroup using Pipe

```
This example uses Get-PvsAuthGroup that returns the PvsAuthGroup named
                theAuthGroup that is piped to Remove-PvsAuthGroup for
                removal.

Get-PvsAuthGroup -Name theAuthGroup | Remove-PvsAuthGroup
```

EXAMPLE 3: Remove All PvsAuthGroup in a Collection Even When Still Used

```
This example removes all PvsAuthGroup in the Collection named
                theCollection of the Site named theSite even when the
                PvsAuthGroup is still being used.

Remove-PvsAuthGroup -SiteName theSite -CollectionName theCollection -
                Force

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 4: Remove All Operator PvsAuthGroup in a Collection using Pipe

```
This example uses Get-PvsAuthGroup that returns a list of Role 400
                PvsAuthGroup in the Collection named theCollection of
                the Site named theSite that is piped to Remove-
                PvsAuthGroup for removal.

Get-PvsAuthGroup -SiteName theSite -CollectionName theCollection -
                Fields Role | Where-Object {$_.Role -eq 400} |
                Remove-PvsAuthGroup

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "Where-Object {$_.Role -eq 400}" only includes PvsAuthGroup with
                Role equal to 400 (Collection Operator).
```

Remove-PvsCollection

```
Remove one or more Collections.

    This required

     Guid[] Guid or CollectionId: GUID of the Collection to Delete.

    or this required & resolution

     string[] Name or CollectionName: Name of the Collection to Delete.

    One of these resolutions when needed

     Guid[] SiteId: GUID of the Site.

     string[] SiteName: Name of the Site.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      CollectionId

    Optional
```

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsCollection with Name theCollection

This example removes the PvsCollection named theCollection in the Site named theSite.

```
Remove-PvsCollection -Name theCollection -SiteName theSite
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Remove the PvsCollection with Name theCollection using Pipe

This example uses Get-PvsCollection that returns the PvsCollection named theCollection in the Site named theSite that is piped to Remove-PvsCollection for removal.

```
Get-PvsCollection -Name theCollection -SiteName theSite | Remove-PvsCollection
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Remove-PvsDevice

Remove one or more Devices.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Delete.

string[] Name or DeviceName: Name of the Device to Delete.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Delete.

Guid[] CollectionId: GUID of the Collection to delete all Devices.

or this required & resolution

string[] CollectionName: Name of the Collection to delete all Devices.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId or CollectionId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsDevice with Name theDevice

```
This example removes the PvsDevice named theDevice.

Remove-PvsDevice -Name theDevice
```

EXAMPLE 2: Remove the PvsDevice with Name theDevice using Pipe

```
This example uses Get-PvsDevice that returns the PvsDevice named
                theDevice that is piped to Remove-PvsDevice for
                removal.

Get-PvsDevice -Name theDevice | Remove-PvsDevice
```

EXAMPLE 3: Remove All PvsDevice in a Collection

```
This example removes all PvsDevice in the Collection named
                theCollection of the Site named theSite.

Remove-PvsDevice -SiteName theSite -CollectionName theCollection

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 4: Remove All Not Active PvsDevice in a Collection using Pipe

```
This example uses Get-PvsDevice that returns a list of not Active
                PvsDevice in the Collection named theCollection of
                the Site named theSite that is piped to Remove-
                PvsDevice for removal.

Get-PvsDevice -SiteName theSite -CollectionName theCollection -Fields
                Active | Where-Object {$_.Active -eq $false} |
                Remove-PvsDevice

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "Where-Object {$_.Active -eq $false}" only includes PvsDevice with
                Active equal to false.
```

Remove-PvsDeviceDiskCacheFile

```
Remove one or more Disk cache files for Devices.

    One of these required

     Guid[] Guid or DeviceId: GUID of the Device to Delete Disk cache
                files.

     string[] Name or DeviceName: Name of the Device to Delete Disk cache
                files.

     PvsPhysicalAddress[] DeviceMac: MAC of the Device to Delete Disk cache
                files.

    This required

     Guid[] DiskLocatorId: GUID of the Disk Locator to Delete Disk cache
                files.

    or this required & resolution
```

string[] DiskLocatorName: Name of the Disk Locator File to Delete Disk cache files.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId or DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsDeviceDiskCacheFile for Device with Name theDevice

This example removes the PvsDeviceDiskCacheFile for the Device named theDevice.

```
Remove-PvsDeviceDiskCacheFile -Name theDevice
```

EXAMPLE 2: Remove the PvsDeviceDiskCacheFile for the Devices that use the DiskLocator with Name theDiskLocator

This example removes the PvsDeviceDiskCacheFile for the Devices that use DiskLocator named theDiskLocator in Site named theSite and Store named theStore.

```
Remove-PvsDeviceDiskCacheFile -DiskLocatorName theDiskLocator -
SiteName theSite -StoreName theStore
```

DiskLocatorId can be used instead of DiskLocatorName so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Remove-PvsDeviceFromDomain

Remove a Device, all Devices in a Collection or View from a Domain.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Remove from the Domain.

string[] Name or DeviceName: Name of the Device to Remove from the Domain.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Remove from the Domain.

Guid[] CollectionId: GUID of the Collection to Remove all Devices from the Domain.

Guid[] SiteViewId: GUID of the Site View to Remove all Devices from the Domain.

Guid[] FarmViewId: GUID of the Farm View to Remove all Devices from the Domain.

string[] FarmViewName: Name of the Farm View toRemove all Devices from the Domain.

or one of these required & resolutions

string[] CollectionName: Name of the Collection to Remove all Devices from the Domain.

string[] SiteViewName: Name of the Site View to Remove all Devices from the Domain.

Optional

string[] Domain: Domain to remove the Device(s) from. If not included, the first Domain Controller found on the Server is used.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId, CollectionId, SiteViewId or FarmViewId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove-PvsDeviceFromDomain for Name

```
Remove-PvsDeviceFromDomain -Name theDevice
```

EXAMPLE 2: Remove-PvsDeviceFromDomain for Name with Domain

```
Remove-PvsDeviceFromDomain -Name theDevice -Domain theDomain
```

EXAMPLE 3: Remove-PvsDeviceFromDomain for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Remove-PvsDeviceFromDomain.

```
Get-PvsDevice -Name theDevice -Fields Guid | Remove-
PvsDeviceFromDomain -Domain theDomain -
OrganizationUnit theOrganizationUnit
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 4: Remove-PvsDeviceFromDomain for FarmViewName

```
Remove-PvsDeviceFromDomain -FarmViewName theFarmView
```

EXAMPLE 5: Remove-PvsDeviceFromDomain for FarmViewName with Domain

```
Remove-PvsDeviceFromDomain -FarmViewName theFarmView -Domain theDomain
```

EXAMPLE 6: Remove-PvsDeviceFromDomain for PvsFarmView Using Pipe

```
The Get-PvsFarmView output is piped to the Remove-PvsDeviceFromDomain.

Get-PvsFarmView -Name theFarmView -Fields Guid | Remove-
                PvsDeviceFromDomain -Domain theDomain -
                OrganizationUnit theOrganizationUnit

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 7: Remove-PvsDeviceFromDomain for CollectionName

```
Remove-PvsDeviceFromDomain -CollectionName theCollection -SiteName
                theSite

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 8: Remove-PvsDeviceFromDomain for CollectionName with Domain

```
Remove-PvsDeviceFromDomain -CollectionName theCollection -SiteName
                theSite -Domain theDomain

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 9: Remove-PvsDeviceFromDomain for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Remove-
                PvsDeviceFromDomain.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                Remove-PvsDeviceFromDomain -Domain theDomain -
                OrganizationUnit theOrganizationUnit

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

EXAMPLE 10: Remove-PvsDeviceFromDomain for SiteViewName

```
Remove-PvsDeviceFromDomain -SiteViewName theSiteView -SiteName theSite

SiteViewId can be used instead of SiteViewName so that the SiteName or
                SiteId is not also needed.
```

EXAMPLE 11: Remove-PvsDeviceFromDomain for SiteViewName with Domain

```
Remove-PvsDeviceFromDomain -SiteViewName theSiteView -SiteName theSite
                -Domain theDomain

SiteViewId can be used instead of SiteViewName so that the SiteName or
                SiteId is not also needed.
```

EXAMPLE 12: Remove-PvsDeviceFromDomain for PvsSiteView Using Pipe

```
The Get-PvsSiteView output is piped to the Remove-PvsDeviceFromDomain.

Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
                Remove-PvsDeviceFromDomain -Domain theDomain -
                OrganizationUnit theOrganizationUnit

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

Remove-PvsDeviceFromView

```
Remove a Device from a View.

    One of these required

     Guid[] Guid or DeviceId: GUID of the Device to Remove.

     string[] Name or DeviceName: Name of the Device to Remove.

     PvsPhysicalAddress[] DeviceMac: MAC of the Device to Remove.

    One of these required

     Guid[] SiteViewId: GUID of the Site View to Remove the Devices from.

     Guid[] FarmViewId: GUID of the Farm View to Remove the Devices from.

     string[] FarmViewName: Name of the Farm View to Remove the Devices
                from.

    or this required & resolution

     string[] SiteViewName: Name of the Site View to Remove the Devices
                from.

    One of these resolutions when needed

     Guid[] SiteId: GUID of the Site.

     string[] SiteName: Name of the Site.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      DeviceId, SiteViewId or FarmViewId

    Optional

     SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "low" to
                have confirmation without the Confirm parameter.
```

EXAMPLE 1: Remove-PvsDeviceFromView for PvsDevice to PvsFarmView

```
Remove-PvsDeviceFromView -Name theDevice -PvsFarmViewName
                thePvsFarmView
```

EXAMPLE 2: Remove-PvsDeviceFromView for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Remove-PvsDeviceFromView.

```
Get-PvsDevice -Name theDevice -Fields Guid | Remove-PvsDeviceFromView
                -PvsFarmViewName thePvsFarmView
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Remove-PvsDeviceFromView for PvsFarmView Using Pipe

The Get-PvsFarmView output is piped to the Remove-PvsDeviceFromView.

```
Get-PvsFarmView -Name theFarmView -Fields Guid | Remove-
                PvsDeviceFromView -Name theDevice
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 4: Remove-PvsDeviceFromView for PvsDevice to PvsSiteView

```
Remove-PvsDeviceFromView -Name theDevice -SiteViewName theSiteView -
                SiteName theSite
```

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

EXAMPLE 5: Remove-PvsDeviceFromView for PvsDevice Using Pipe

The Get-PvsDevice output is piped to the Remove-PvsDeviceFromView.

```
Get-PvsDevice -Name theDevice -Fields Guid | Remove-PvsDeviceFromView
                -SiteViewName theSiteView -SiteName theSite
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

EXAMPLE 6: Remove-PvsDeviceFromView for PvsSiteView Using Pipe

The Get-PvsSiteView output is piped to the Remove-PvsDeviceFromView.

```
Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
                Remove-PvsDeviceFromView -Name theDevice
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

Remove-PvsDirectory

Remove a Directory on the Server specified.

One of these required

Guid[] Guid or ServerId: GUID of the Server to remove a Directory from. The directory must be empty to be removed.

string[] Name or ServerName: Name of the Server to remove a Directory from. The directory must be empty to be removed.

This required

string[] Path: Path of the Directory to remove.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove-PvsDirectory for Name

```
Remove-PvsDirectory -Name theServer -Path "C:\directory\subdirectory"
```

EXAMPLE 2: Remove-PvsDirectory for PvsServer Using Pipe

The Get-PvsServer output is piped to the Remove-PvsDirectory.

```
Get-PvsServer -Name theServer -Fields Guid | Remove-PvsDirectory -Path "C:\directory\subdirectory"
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Remove-PvsDiskFromUpdateTask

Remove a Disk from an Update Task.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Remove.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator File to Remove.

One of these optional

Guid[] UpdateTaskId: GUID of the Update Task to Remove a Disk.

string[] UpdateTaskName: Name of the Update Task to Remove a Disk.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId or UpdateTaskId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove-DiskFromUpdateTask for PvsDiskLocator to PvsUpdateTask

```
Remove-DiskFromUpdateTask -Name theDiskLocator -UpdateTaskName
theUpdateTask -SiteName theSite -StoreName theStore
```

UpdateTaskId can be used instead of UpdateTaskName so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Remove-DiskFromUpdateTask for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Remove-DiskFromUpdateTask.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Fields Guid | Remove-DiskFromUpdateTask -
UpdateTaskName theUpdateTask -SiteName theSite
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Remove-DiskFromUpdateTask for PvsUpdateTask Using Pipe

The Get-PvsUpdateTask output is piped to the Remove-DiskFromUpdateTask.

```
Get-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Fields Guid |
Remove-DiskFromUpdateTask -Name theDiskLocator -
SiteName theSite -StoreName theStore
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Remove-PvsDiskLocator

Remove one or more Disk Locators.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Delete.

or one of these required & resolutions

string[] Name or DiskLocatorName: Name of the Disk Locator File to Delete.

Guid[] StoreId: GUID of the Store to delete all DiskLocators.

string[] StoreName: Name of the Store to delete all DiskLocators.

Optional

SwitchParameter DeleteDiskFile: If -DiskFile is specified, the Disk File will be deleted.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store to delete all DiskLocators.

string[] StoreName: Name of the Store to delete all DiskLocators.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId or StoreId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsDiskLocator with Name theDiskLocator

This example removes the PvsDiskLocator named theDiskLocator in the Site named theSite and Store named theStore and deletes the vDisk file.

```
Remove-PvsDiskLocator -Name theDiskLocator -SiteName theSite -
StoreName theStore -DeleteDiskFile
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 2: Remove the PvsDiskLocator with Name theDiskLocator using Pipe

This example uses Get-PvsDiskLocator that returns the PvsDiskLocator named theDiskLocator in the Site named theSite and Store named theStore that is piped to Remove-PvsDiskLocator for removal.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore | Remove-PvsDiskLocator
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Remove-PvsDiskLocatorFromDevice

Remove a Disk Locator from a Device, Collection, View, or Site.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Remove.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator File to Remove.

One of these optional

Guid[] DeviceId: GUID of the Device to Remove a Disk Locator.

string[] DeviceName: Name of the Device to Remove a Disk Locator.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Remove a Disk Locator.

Guid[] CollectionId: GUID of the Collection to Remove a Disk Locator or Locators from all Devices.

Guid[] SiteViewId: GUID of the Site View to Remove a Disk Locator from all Devices.

Guid[] FarmViewId: GUID of the Farm View to Remove a Disk Locator from all Devices.

string[] FarmViewName: Name of the Farm View to Remove a Disk Locator from all Devices.

or one of these optional & resolutions

string[] CollectionName: Name of the Collection to Remove a Disk Locator or Locators from all Devices.

string[] SiteViewName: Name of the Site View to Remove a Disk Locator from all Devices.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId, DeviceId, CollectionId, SiteViewId or FarmViewId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove-PvsDiskLocatorFromDevice for PvsDiskLocator to PvsDevice

```
Remove-PvsDiskLocatorFromDevice -Name theDiskLocator -DeviceName
theDevice -SiteName theSite -StoreName theStore
```

```
Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 2: Remove-PvsDiskLocatorFromDevice for PvsDiskLocator Using Pipe

```
The Get-PvsDiskLocator output is piped to the Remove-
                PvsDiskLocatorFromDevice.

Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Remove-
                PvsDiskLocatorFromDevice -DeviceName theDevice

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 3: Remove-PvsDiskLocatorFromDevice for PvsDevice Using Pipe

```
The Get-PvsDevice output is piped to the Remove-
                PvsDiskLocatorFromDevice.

Get-PvsDevice -Name theDevice -Fields Guid | Remove-
                PvsDiskLocatorFromDevice -Name theDiskLocator -
                SiteName theSite -StoreName theStoreName

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 4: Remove-PvsDiskLocatorFromDevice for PvsDiskLocator to PvsCollection

```
Remove-PvsDiskLocatorFromDevice -Name theDiskLocator -CollectionName
                theCollection -SiteName theSite -StoreName theStore
```

EXAMPLE 5: Remove-PvsDiskLocatorFromDevice for PvsDiskLocator Using Pipe

```
The Get-PvsDiskLocator output is piped to the Remove-
                PvsDiskLocatorFromDevice.

Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Remove-
                PvsDiskLocatorFromDevice -CollectionName
                theCollection -SiteName theSite

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 6: Remove-PvsDiskLocatorFromDevice for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Remove-
                PvsDiskLocatorFromDevice.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                Remove-PvsDiskLocatorFromDevice -Name theDiskLocator
                -SiteName theSite -StoreName theStoreName
```

```
    The -Fields parameter with only the needed fields specified makes the
                    Get work faster because only those fields are
                    retrieved.
```

EXAMPLE 7: Remove-PvsDiskLocatorFromDevice for PvsDiskLocator to PvsFarmView

```
    Remove-PvsDiskLocatorFromDevice -Name theDiskLocator -FarmViewName
                    theFarmView -SiteName theSite -StoreName theStore

    Guid can be used instead of Name so that the SiteName or SiteId and
                    StoreName or StoreId are not also needed.
```

EXAMPLE 8: Remove-PvsDiskLocatorFromDevice for PvsDiskLocator Using Pipe

```
    The Get-PvsDiskLocator output is piped to the Remove-
                    PvsDiskLocatorFromDevice.

    Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                    theStore -Fields Guid | Remove-
                    PvsDiskLocatorFromDevice -FarmViewName theFarmView

    The -Fields parameter with only the needed fields specified makes the
                    Get work faster because only those fields are
                    retrieved.

    Guid can be used instead of Name so that the SiteName or SiteId and
                    StoreName or StoreId are not also needed.
```

EXAMPLE 9: Remove-PvsDiskLocatorFromDevice for PvsFarmView Using Pipe

```
    The Get-PvsFarmView output is piped to the Remove-
                    PvsDiskLocatorFromDevice.

    Get-PvsFarmView -Name theFarmView -Fields Guid | Remove-
                    PvsDiskLocatorFromDevice -Name theDiskLocator -
                    SiteName theSite -StoreName theStoreName

    The -Fields parameter with only the needed fields specified makes the
                    Get work faster because only those fields are
                    retrieved.

    Guid can be used instead of Name so that the SiteName or SiteId and
                    StoreName or StoreId are not also needed.
```

EXAMPLE 10: Remove-PvsDiskLocatorFromDevice for PvsDiskLocator to PvsSiteView

```
    Remove-PvsDiskLocatorFromDevice -Name theDiskLocator -SiteViewName
                    theSiteView -SiteName theSite -StoreName theStore
```

EXAMPLE 11: Remove-PvsDiskLocatorFromDevice for PvsDiskLocator Using Pipe

```
    The Get-PvsDiskLocator output is piped to the Remove-
                    PvsDiskLocatorFromDevice.

    Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                    theStore -Fields Guid | Remove-
                    PvsDiskLocatorFromDevice -SiteViewName theSiteView -
                    SiteName theSite

    The -Fields parameter with only the needed fields specified makes the
                    Get work faster because only those fields are
                    retrieved.
```

EXAMPLE 12: Remove-PvsDiskLocatorFromDevice for PvsSiteView Using Pipe

```
The Get-PvsSiteView output is piped to the Remove-
                PvsDiskLocatorFromDevice.

Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
                Remove-PvsDiskLocatorFromDevice -Name theDiskLocator
                -SiteName theSite -StoreName theStoreName

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

Remove-PvsDiskUpdateDevice

```
Remove one or more Disk Update Devices.

    One of these required

     Guid[] Guid or DeviceId: GUID of the Disk Update Device to Delete.

     string[] Name or DeviceName: Name of the Disk Update Device to Delete.

     PvsPhysicalAddress[] DeviceMac: MAC of the Disk Update Device to
                Delete.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      DeviceId

    Optional

     SwitchParameter Confirm: The impact of this operation is "medium". If
                -Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "medium"
                or "low" to have confirmation without the Confirm
                parameter.
```

EXAMPLE 1: Remove the PvsDiskUpdateDevice with Name theDevice

```
This example removes the PvsDiskUpdateDevice named theDevice.

Remove-PvsDiskUpdateDevice -Name theDevice
```

EXAMPLE 2: Remove the PvsDiskUpdateDevice with Name theDevice using Pipe

```
This example uses Get-PvsDiskUpdateDevice that returns the
                PvsDiskUpdateDevice named theDevice that is piped to
                Remove-PvsDiskUpdateDevice for removal.

Get-PvsDiskUpdateDevice -Name theDevice | Remove-PvsDiskUpdateDevice
```

EXAMPLE 3: Remove All Not Active PvsDiskUpdateDevice for a DiskLocator using Pipe

```
This example uses Get-PvsDiskUpdateDevice that returns a list of not
                Active PvsDiskUpdateDevice for the DiskLocator named
                theDiskLocator of the Site named theSite and Store
                named theStore that is piped to Remove-
                PvsDiskUpdateDevice for removal.

Get-PvsDiskUpdateDevice -DiskLocatorName theDiskLocator -SiteName
                theSite -StoreName theStore -Fields Active | Where-
```

Object {$_.Active -eq $false} | Remove-PvsDiskUpdateDevice

DiskLocatorId can be used instead of DiskLocatorName so that the SiteName or SiteId and StoreName or StoreId are not also needed.

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "Where-Object {$_.Active -eq $false}" only includes PvsDiskUpdateDevice with Active equal to false.

Remove-PvsDiskVersion

Remove the latest Disk version or no longer needed version if no Devices are currently booted from that version.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to Delete the Version from.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator File to Delete the Version from.

Optional

uint Version: Specifies the version that should be deleted. Used when deleting versions that are no longer needed because of a Merge.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

(DiskLocatorId and Version) or DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the Highest PvsDiskVersion with Name theDiskLocator

```
This example removes the highest PvsDiskVersion for DiskLocator named
                theDiskLocator in the Site named theSite and Store
                named theStore.

Remove-PvsDiskVersion -Name theDiskLocator -SiteName theSite -
                StoreName theStore

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 2: Remove the highest PvsDiskVersion with Name theDiskLocator using Pipe

```
This example uses Get-PvsDiskLocator that returns the PvsDiskLocator
                named theDiskLocator in the Site named theSite and
                Store named theStore that is piped to Remove-
                PvsDiskVersion for removal of the highest version.

Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore | Remove-PvsDiskVersion

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 3: Remove the Version PvsDiskVersion with Name theDiskLocator

```
This example removes the Version PvsDiskVersion for DiskLocator named
                theDiskLocator in the Site named theSite and Store
                named theStore.

Remove-PvsDiskVersion -Name theDiskLocator -SiteName theSite -
                StoreName theStore -Version 5

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 4: Remove the Version PvsDiskVersion with Name theDiskLocator using Pipe

```
This example uses Get-PvsDiskVersion that returns the Version
                PvsDiskVersion for DiskLocator named theDiskLocator
                in the Site named theSite and Store named theStore
                that is piped to Remove-PvsDiskVersion for removal of
                the specified version.

Get-PvsDiskVersion -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Version 5 | Remove-PvsDiskVersion

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

Remove-PvsFarmView

```
Remove one or more Views from the Farm.

    One of these required

     Guid[] Guid or FarmViewId: GUID of the Farm View to Delete.

     string[] Name or FarmViewName: Name of the Farm View to Delete.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:
```

FarmViewId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsFarmView with Name theFarmView

This example removes the PvsFarmView named theFarmView.

```
Remove-PvsFarmView -Name theFarmView
```

EXAMPLE 2: Remove the PvsFarmView with Name theFarmView using Pipe

This example uses Get-PvsFarmView that returns the PvsFarmView named theFarmView that is piped to Remove-PvsFarmView for removal.

```
Get-PvsFarmView -Name theFarmView | Remove-PvsFarmView
```

Remove-PvsServer

Remove one or more Servers.

One of these required

Guid[] Guid or ServerId: GUID of the Server to Delete.

string[] Name or ServerName: Name of the Server to Delete.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsServer with Name theServer

This example removes the PvsServer named theServer.

```
Remove-PvsServer -Name theServer
```

EXAMPLE 2: Remove the PvsServer with Name theServer using Pipe

This example uses Get-PvsServer that returns the PvsServer named theServer that is piped to Remove-PvsServer for removal.

```
Get-PvsServer -Name theServer | Remove-PvsServer
```

Remove-PvsServerStore

Remove the connection from Servers to Stores.

One of these required

Guid[] Guid or ServerId: GUID of a Server that uses the path to get to the Store.

string[] Name or ServerName: Name of a Server that uses the path to get to the Store.

One of these required

Guid[] StoreId: GUID of the Store.

string[] StoreName: Name of the Store.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId or StoreId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsServerStore with Name theServer/theStore

This example removes the PvsServerStore for the Server named theServer and Store named theStore.

```
Remove-PvsServerStore -Name theServer -StoreName theStore
```

EXAMPLE 2: Remove the PvsServerStore with Name theServer/theStore using Pipe

This example uses Get-PvsServer that returns the PvsServer named theServer that is piped to Remove-PvsServerStore with StoreName theStore for removal.

```
Get-PvsServer -Name theServer | Remove-PvsServerStore -StoreName theStore
```

Remove-PvsSite

Remove one or more Sites.

One of these required

Guid[] Guid or SiteId: GUID of the Site to Delete.

string[] Name or SiteName: Name of the Site to Delete.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

SiteId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium"

or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsSite with Name theSite

This example removes the PvsSite named theSite.

```
Remove-PvsSite -Name theSite
```

EXAMPLE 2: Remove the PvsSite with Name theSite using Pipe

This example uses Get-PvsSite that returns the PvsSite named theSite that is piped to Remove-PvsSite for removal.

```
Get-PvsSite -Name theSite | Remove-PvsSite
```

Remove-PvsSiteView

Remove one or more Views from Sites.

This required

Guid[] Guid or SiteViewId: GUID of the Site View to Delete.

or this required & resolution

string[] Name or SiteViewName: Name of the Site View to Delete.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

SiteViewId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsSiteView with Name theSiteView

This example removes the PvsSiteView named theSiteView in the Site named theSite.

```
Remove-PvsSiteView -Name theSiteView -SiteName theSite
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Remove the PvsSiteView with Name theSiteView using Pipe

This example uses Get-PvsSiteView that returns the PvsSiteView named theSiteView in the Site named theSite that is piped to Remove-PvsSiteView for removal.

```
Get-PvsSiteView -Name theSiteView -SiteName theSite | Remove-
PvsSiteView
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Remove-PvsStore

Remove one or more Stores.

One of these required

Guid[] Guid or StoreId: GUID of the Store to Delete.

string[] Name or StoreName: Name of the Store to Delete.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

StoreId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsStore with Name theStore

This example removes the PvsStore named theStore.

```
Remove-PvsStore -Name theStore
```

EXAMPLE 2: Remove the PvsStore with Name theStore using Pipe

This example uses Get-PvsStore that returns the PvsStore named theStore that is piped to Remove-PvsStore for removal.

```
Get-PvsStore -Name theStore | Remove-PvsStore
```

Remove-PvsUpdateTask

Remove one or more Update Tasks from Sites.

This required

Guid[] Guid or UpdateTaskId: GUID of the Update Task to Delete.

or this required & resolution

string[] Name or UpdateTaskName: Name of the Update Task to Delete.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

UpdateTaskId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsUpdateTask with Name theUpdateTask

This example removes the PvsUpdateTask named theUpdateTask in the Site named theSite.

```
Remove-PvsUpdateTask -Name theUpdateTask -SiteName theSite
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Remove the PvsUpdateTask with Name theUpdateTask using Pipe

This example uses Get-PvsUpdateTask that returns the PvsUpdateTask named theUpdateTask in the Site named theSite that is piped to Remove-PvsUpdateTask for removal.

```
Get-PvsUpdateTask -Name theUpdateTask -SiteName theSite | Remove-PvsUpdateTask
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Remove-PvsVirtualHostingPool

Remove one or more Virtual Hosting Pools from Sites.

This required

Guid[] Guid or VirtualHostingPoolId: GUID of the Virtual Hosting Pool to Delete.

or this required & resolution

string[] Name or VirtualHostingPoolName: Name of the Virtual Hosting Pool to Delete.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

VirtualHostingPoolId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Remove the PvsVirtualHostingPool with Name theVirtualHostingPool

```
This example removes the PvsVirtualHostingPool named
                theVirtualHostingPool in the Site named theSite.

Remove-PvsVirtualHostingPool -Name theVirtualHostingPool -SiteName
                theSite

Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

EXAMPLE 2: Remove the PvsVirtualHostingPool with Name theVirtualHostingPool using Pipe

```
This example uses Get-PvsVirtualHostingPool that returns the
                PvsVirtualHostingPool named theVirtualHostingPool in
                the Site named theSite that is piped to Remove-
                PvsVirtualHostingPool for removal.

Get-PvsVirtualHostingPool -Name theVirtualHostingPool -SiteName
                theSite | Remove-PvsVirtualHostingPool

Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

Reset-PvsDatabase

Cause the database location to be reloaded.

EXAMPLE 1: Reset-PvsDatabase

```
Reset-PvsDatabase
```

Reset-PvsDeviceForDomain

Reset a Device, all Devices in a Collection or View for a Domain.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Reset for the Domain.

string[] Name or DeviceName: Name of the Device to Reset for the Domain.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Reset for the Domain.

Guid[] CollectionId: GUID of the Collection to Reset all Devices for the Domain.

Guid[] SiteViewId: GUID of the Site View to Reset all Devices for the Domain.

Guid[] FarmViewId: GUID of the Farm View to Reset all Devices for the Domain.

string[] FarmViewName: Name of the Farm View to Reset all Devices for the Domain.

or one of these required & resolutions

string[] CollectionName: Name of the Collection to Reset all Devices for the Domain.

string[] SiteViewName: Name of the Site View to Reset all Devices for the Domain.

```
Optional

 string[] Domain: Domain to Reset the Device(s) for. If not included,
                  the first Domain Controller found on the Server is
                  used.

 string[] OrganizationUnit: Organizational Unit to reset the Device(s)
                  to. This parameter is optional. If it is not
                  specified, the account remains in its existing OU.
                  Child OU's should be delimited with forward slashes,
                  e.g. "ParentOU/ChildOU". Special characters in an OU
                  name, such as '"', '#', '+', ',', ';', '>', '=', must
                  be escaped with a backslash. For example, an OU
                  called "commaIn,TheMiddle" must be specified as
                  "commaIn\,TheMiddle". The old syntax of delimiting
                  child OU's with a comma is still supported, but
                  deprecated. Note that in this case, the child OU
                  comes first, e.g. "ChildOU,ParentOU".

One of these resolutions when needed

 Guid[] SiteId: GUID of the Site.

 string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

 PvsObject[] Object: PvsObjects with the members below can be used as
                  the Object parameter or from a pileline:

  DeviceId, CollectionId, SiteViewId or FarmViewId

Optional

 SwitchParameter Confirm: The impact of this operation is "medium". If
                  -Confirm is specified, the operation will be
                  confirmed. $ConfirmPreference can be set to "medium"
                  or "low" to have confirmation without the Confirm
                  parameter.
```

EXAMPLE 1: Reset-PvsDeviceForDomain for Name

```
Reset-PvsDeviceForDomain -Name theDevice
```

EXAMPLE 2: Reset-PvsDeviceForDomain for Name with Domain

```
Reset-PvsDeviceForDomain -Name theDevice -Domain theDomain
```

EXAMPLE 3: Reset-PvsDeviceForDomain for Name with Domain and OrganizationUnit

```
Reset-PvsDeviceForDomain -Name theDevice -Domain theDomain -
                  OrganizationUnit theOrganizationUnit
```

EXAMPLE 4: Reset-PvsDeviceForDomain for PvsDevice Using Pipe

```
The Get-PvsDevice output is piped to the Reset-PvsDeviceForDomain.

Get-PvsDevice -Name theDevice -Fields Guid | Reset-PvsDeviceForDomain
                  -Domain theDomain -OrganizationUnit
                  theOrganizationUnit

The -Fields parameter with only the needed fields specified makes the
                  Get work faster because only those fields are
                  retrieved.
```

EXAMPLE 5: Reset-PvsDeviceForDomain for FarmViewName

```
Reset-PvsDeviceForDomain -FarmViewName theFarmView
```

EXAMPLE 6: Reset-PvsDeviceForDomain for FarmViewName with Domain

```
Reset-PvsDeviceForDomain -FarmViewName theFarmView -Domain theDomain
```

EXAMPLE 7: Reset-PvsDeviceForDomain for FarmViewName with Domain and OrganizationUnit

```
Reset-PvsDeviceForDomain -FarmViewName theFarmView -Domain theDomain -
                OrganizationUnit theOrganizationUnit
```

EXAMPLE 8: Reset-PvsDeviceForDomain for PvsFarmView Using Pipe

```
The Get-PvsFarmView output is piped to the Reset-PvsDeviceForDomain.

Get-PvsFarmView -Name theFarmView -Fields Guid | Reset-
                PvsDeviceForDomain -Domain theDomain -
                OrganizationUnit theOrganizationUnit

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 9: Reset-PvsDeviceForDomain for CollectionName

```
Reset-PvsDeviceForDomain -CollectionName theCollection -SiteName
                theSite

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 10: Reset-PvsDeviceForDomain for CollectionName with Domain

```
Reset-PvsDeviceForDomain -CollectionName theCollection -SiteName
                theSite -Domain theDomain

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 11: Reset-PvsDeviceForDomain for CollectionName with Domain and OrganizationUnit

```
Reset-PvsDeviceForDomain -CollectionName theCollection -SiteName
                theSite -Domain theDomain -OrganizationUnit
                theOrganizationUnit

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 12: Reset-PvsDeviceForDomain for PvsCollection Using Pipe

```
The Get-PvsCollection output is piped to the Reset-PvsDeviceForDomain.

Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
                Reset-PvsDeviceForDomain -Domain theDomain -
                OrganizationUnit theOrganizationUnit

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

EXAMPLE 13: Reset-PvsDeviceForDomain for SiteViewName

```
Reset-PvsDeviceForDomain -SiteViewName theSiteView -SiteName theSite
```

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

EXAMPLE 14: Reset-PvsDeviceForDomain for SiteViewName with Domain

```
Reset-PvsDeviceForDomain -SiteViewName theSiteView -SiteName theSite -
Domain theDomain
```

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

EXAMPLE 15: Reset-PvsDeviceForDomain for SiteViewName with Domain and OrganizationUnit

```
Reset-PvsDeviceForDomain -SiteViewName theSiteView -SiteName theSite -
Domain theDomain -OrganizationUnit
theOrganizationUnit
```

SiteViewId can be used instead of SiteViewName so that the SiteName or SiteId is not also needed.

EXAMPLE 16: Reset-PvsDeviceForDomain for PvsSiteView Using Pipe

The Get-PvsSiteView output is piped to the Reset-PvsDeviceForDomain.

```
Get-PvsSiteView -Name theSiteView -SiteName theSite -Fields Guid |
Reset-PvsDeviceForDomain -Domain theDomain -
OrganizationUnit theOrganizationUnit
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Restart-PvsStreamService

Restart the Stream Service on a Server or all Servers in a Site.

One of these required

Guid[] Guid or ServerId: GUID of the Server to restart the Stream Service.

string[] Name or ServerName: Name of the Server to restart the Stream Service.

Guid[] SiteId: GUID of the Site to restart the Stream Service on all Servers.

string[] SiteName: Name of the Site to restart the Stream Service on all Servers.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId or SiteId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Restart-PvsStreamService for Name

```
Restart-PvsStreamService -Name theServer
```

EXAMPLE 2: Restart-PvsStreamService for PvsServer Using Pipe

The Get-PvsServer output is piped to the Restart-PvsStreamService.

```
Get-PvsServer -Name theServer -Fields Guid | Restart-PvsStreamService
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Restart-PvsStreamService for SiteName

```
Restart-PvsStreamService -SiteName theSite
```

EXAMPLE 4: Restart-PvsStreamService for PvsSite Using Pipe

The Get-PvsSite output is piped to the Restart-PvsStreamService.

```
Get-PvsSite -Name theSite -Fields Guid | Restart-PvsStreamService
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Revoke-PvsAuthGroup

Remove Farm, Site or Collection Authorization for an AuthGroup. If no Site or Collection is specified, Farm Authorization is removed for the AuthGroup.

One of these required

Guid[] Guid or AuthGroupId: GUID of the AuthGroup to remove Authorization for.

string[] Name or AuthGroupName: Name of the AuthGroup to remove Authorization for.

One of these optional

Guid[] SiteId: GUID of the Site to remove Authorization for the AuthGroup.

string[] SiteName: Name of the Site to remove Authorization for the AuthGroup.

Guid[] CollectionId: GUID of the Collection to remove Authorization for the AuthGroup.

or this optional & resolution

string[] CollectionName: Name of the Collection to remove Authorization for the AuthGroup.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site to remove Authorization for the AuthGroup.

string[] SiteName: Name of the Site to remove Authorization for the AuthGroup.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

AuthGroupId, SiteId or CollectionId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Revoke-PvsAuthGroup for PvsAuthGroup to PvsFarm

```
Revoke-PvsAuthGroup -Name theAuthGroup
```

EXAMPLE 2: Revoke-PvsAuthGroup for PvsAuthGroup Using Pipe

The Get-PvsAuthGroup output is piped to the Revoke-PvsAuthGroup.

```
Get-PvsAuthGroup -Name theAuthGroup -Fields Guid | Revoke-PvsAuthGroup
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Revoke-PvsAuthGroup for PvsAuthGroup to PvsSite

```
Revoke-PvsAuthGroup -Name theAuthGroup -SiteName theSite
```

EXAMPLE 4: Revoke-PvsAuthGroup for PvsAuthGroup Using Pipe

The Get-PvsAuthGroup output is piped to the Revoke-PvsAuthGroup.

```
Get-PvsAuthGroup -Name theAuthGroup -Fields Guid | Revoke-PvsAuthGroup
-SiteName theSite
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 5: Revoke-PvsAuthGroup for PvsSite Using Pipe

The Get-PvsSite output is piped to the Revoke-PvsAuthGroup.

```
Get-PvsSite -Name theSite -Fields Guid | Revoke-PvsAuthGroup -Name
theAuthGroup
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 6: Revoke-PvsAuthGroup for PvsAuthGroup to PvsCollection

```
Revoke-PvsAuthGroup -Name theAuthGroup -CollectionName theCollection -
SiteName theSite
```

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 7: Revoke-PvsAuthGroup for PvsAuthGroup Using Pipe

The Get-PvsAuthGroup output is piped to the Revoke-PvsAuthGroup.

```
Get-PvsAuthGroup -Name theAuthGroup -Fields Guid | Revoke-PvsAuthGroup
    -CollectionName theCollection -SiteName theSite
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 8: Revoke-PvsAuthGroup for PvsCollection Using Pipe

The Get-PvsCollection output is piped to the Revoke-PvsAuthGroup.

```
Get-PvsCollection -Name theCollection -SiteName theSite -Fields Guid |
    Revoke-PvsAuthGroup -Name theAuthGroup
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Set-PvsAuthGroup

Set AuthGroup(s) changed values from PvsAuthGroup object(s), or set one or more field values for a PvsAuthGroup.

Required

PvsAuthGroup AuthGroup: PvsAuthGroup object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsAuthGroup object, and only will be set if the value has changed.

string Name or AuthGroupName: Name of the Active Directory or Windows Group. Max Length=450

string Description: User description. Default="" Max Length=250

When AuthGroup is not passed, the parameters below are used:

One of these required

Guid Guid or AuthGroupId: GUID of the AuthGroup to Set.

string Name or AuthGroupName: Name of the AuthGroup to Set.

Optional field values to set:

string NewName: Name of the Active Directory or Windows Group. Max Length=450

string Description: User description. Default="" Max Length=250

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsAuthGroup object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsAuthGroup for Individual Fields

Get the PvsAuthGroup into a $o variable. Change the $o field values and then Set the PvsAuthGroup with the result.

```
$o = Get-PvsAuthGroup -Name oldName -Fields Name
```

```
$o.Name = "newName"
```

```
Set-PvsAuthGroup $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsAuthGroup for a Field Using Pipe

Get the PvsAuthGroup into a $o variable. Change a $o field to the correct value and then Set the PvsAuthGroup with the result.

```
Get-PvsAuthGroup -Name oldName -Fields Name | foreach { $o = $_; $o.Name = "newName"; $o } | Set-PvsAuthGroup
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsCeipData

Set Ceip changed values from a PvsCeip object, or set one or more field values for a PvsCeip.

Required

PvsCeipData CeipData: PvsCeipData object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsCeipData object, and only will be set if the value has changed.

uint Enabled: 1 if CEIP is enabled, otherwise 0. Min=0, Max=1

DateTime NextUpload: Date and time next CEIP upload is due if enabled is 1. Default=Empty

uint InProgress: 1 if an upload is currently in progress, otherwise 0. Default=0

```
    Guid ServerId: ID of server that is currently uploading, null if
                   InProgress is 0. Default=00000000-0000-0000-0000-
                   000000000000

    uint OneTimeUpload: 1 to perform a one time upload. Default=0


  When CeipData is not passed, the parameters below are used:

  Optional

    string Uuid: CEIP UUID of this Farm. This is optional since there is
                 only one.

  Optional field values to set:

    uint Enabled: 1 if CEIP is enabled, otherwise 0. Min=0, Max=1

    DateTime NextUpload: Date and time next CEIP upload is due if enabled
                 is 1. Default=Empty

    uint InProgress: 1 if an upload is currently in progress, otherwise 0.
                 Default=0

    Guid ServerId: ID of server that is currently uploading, null if
                   InProgress is 0. Default=00000000-0000-0000-0000-
                   000000000000

    uint OneTimeUpload: 1 to perform a one time upload. Default=0


  Optional

    SwitchParameter PassThru: If -PassThru is specified, the resulting
                   PvsCeipData object(s) are returned.

    SwitchParameter Confirm: The impact of this operation is "low". If -
                   Confirm is specified, the operation will be
                   confirmed. $ConfirmPreference can be set to "low" to
                   have confirmation without the Confirm parameter.
```

EXAMPLE 1: Set PvsCeipData for Individual Fields

```
    Get the PvsCeipData into a $o variable. Change the $o field values and
                   then Set the PvsCeipData with the result.

    $o = Get-PvsCeipData -Fields Enabled, NextUpload

    $o.Enabled = $true

    $o.NextUpload = "2016-01-14 15:52:00"

    Set-PvsCeipData $o

    The -Fields parameter with only the needed fields specified makes the
                   Get work faster because only those fields are
                   retrieved.
```

EXAMPLE 2: Set PvsCeipData for a Field Using Pipe

```
    Get the PvsCeipData into a $o variable for the field that has the
                   wrong value. Change the $o field to the correct value
                   and then Set the PvsCeipData with the result.
```

```
Get-PvsCeipData -Fields Enabled | Where-Object {$_.Enabled -ne $false}
                | foreach { $o = $_; $o.Enabled = $false; $o } | Set-
                PvsCeipData
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsCisData

Set one or more field values for CIS data.

Required

PvsCisData CisData: PvsCisData object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsCisData object, and only will be set if the value has changed.

string UserName: Username used to obtain the token Default="" Max Length=255

string Path: Path where the last problem report bundle was saved Default="" Max Length=255

string Password: Password of the user required to obtain the token. This is required only by Set and Add

When CisData is not passed, the parameters below are used:

Optional

Guid Guid or CisDataId: CIS UUID of this Farm. This is optional since there is only a single record.

Optional field values to set:

string UserName: Username used to obtain the token Default="" Max Length=255

string Path: Path where the last problem report bundle was saved Default="" Max Length=255

string Password: Password of the user required to obtain the token. This is required only by Set and Add

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsCisData object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsCisData for Individual Fields

```
Get the PvsCisData into a $o variable. Change the $o field values and
                then Set the PvsCisData with the result.

$o = Get-PvsCisData -Fields UserName, UploadToken

$o.UserName = userName

$o.UploadToken = token

Set-PvsCisData $o

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 2: Set PvsCisData for a Field Using Pipe

```
Get the PvsCisData into a $o variable for the field that has the wrong
                value. Change the $o field to the correct value and
                then Set the PvsCisData with the result.

Get-PvsCisData -Fields UserName | Where-Object {$_.UserName -ne
                newUserName} | foreach { $o = $_; $o.UserName =
                newUserName; $o } | Set-PvsCisData

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y
                and returns the object again so it can be piped to
                the Set command for update.
```

Set-PvsCollection

```
Set Collection(s) changed values from PvsCollection object(s), or set one
   or more field values for a PvsCollection.

    Required

     PvsCollection Collection: PvsCollection object(s) with changed
                property value(s) to be set. The object(s) can come
                from a pileline.

     These values are in the PvsCollection object, and only will be set if
                the value has changed.

     string Name or CollectionName: Name of the Collection. It is unique
                within the Site. Max Length=50

     string Description: User description. Default="" Max Length=250

     Guid TemplateDeviceId: GUID of a Device in the Collection whose
                settings are used for initial values of new Devices.
                Not used with templateDeviceName. Default=00000000-
                0000-0000-0000-000000000000

     string TemplateDeviceName: Name of a Device in the Collection whose
                settings are used for initial values of new Devices.
                Not used with TemplateDeviceId. Default=""

     uint LastAutoAddDeviceNumber: The Device Number of the last Auto Added
                Device. Default=0
```

bool Enabled: True when Devices in the Collection can be booted, false otherwise. Default=true

string AutoAddPrefix: The string put before the Device Number for Auto Add. Default="" ASCII computer name characters no end digit Max Length=12

string AutoAddSuffix: The string put after the Device Number for Auto Add. Default="" ASCII computer name characters no begin digit Max Length=12

bool AutoAddZeroFill: True when zeros be placed before the Device Number up to the AutoAddNumberLength for Auto Add, false otherwise. Default=true

uint AutoAddNumberLength: The maximum length of the Device Number for Auto Add. This length plus the AutoAddPrefix length plus the AutoAddSuffix length must be less than 16. Required that ((lenautoAddPrefix+lenautoAddSuffix)+AutoAddNumberLength)<=15. Min=3, Max=9, Default=4

When Collection is not passed, the parameters below are used:

This required

Guid Guid or CollectionId: GUID of the Collection to Set.

or this required & resolution

string Name or CollectionName: Name of the Collection to Set.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

Optional field values to set:

string NewName: Name of the Collection. It is unique within the Site. Max Length=50

string Description: User description. Default="" Max Length=250

Guid TemplateDeviceId: GUID of a Device in the Collection whose settings are used for initial values of new Devices. Not used with templateDeviceName. Default=00000000-0000-0000-0000-000000000000

string TemplateDeviceName: Name of a Device in the Collection whose settings are used for initial values of new Devices. Not used with TemplateDeviceId. Default=""

uint LastAutoAddDeviceNumber: The Device Number of the last Auto Added Device. Default=0

bool Enabled: True when Devices in the Collection can be booted, false otherwise. Default=true

string AutoAddPrefix: The string put before the Device Number for Auto Add. Default="" ASCII computer name characters no end digit Max Length=12

```
string AutoAddSuffix: The string put after the Device Number for Auto
                Add. Default="" ASCII computer name characters no
                begin digit Max Length=12

bool AutoAddZeroFill: True when zeros be placed before the Device
                Number up to the AutoAddNumberLength for Auto Add,
                false otherwise. Default=true

uint AutoAddNumberLength: The maximum length of the Device Number for
                Auto Add. This length plus the AutoAddPrefix length
                plus the AutoAddSuffix length must be less than 16.
                Required that
                ((lenautoAddPrefix+lenautoAddSuffix)+AutoAddNumberLen
                gth)<=15. Min=3, Max=9, Default=4


Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting
                PvsCollection object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "low" to
                have confirmation without the Confirm parameter.
```

EXAMPLE 1: Set PvsCollection for Individual Fields

```
Get the PvsCollection into a $o variable. Change the $o field values
                and then Set the PvsCollection with the result.

$o = Get-PvsCollection -Name theCollection -SiteName theSite -Fields
                AutoAddSuffix, AutoAddZeroFill

$o.AutoAddSuffix = "Ex"

$o.AutoAddZeroFill = $true

Set-PvsCollection $o

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not
                also needed.
```

EXAMPLE 2: Set PvsCollection for a Field Using Pipe

```
Get the PvsCollection into a $o variable. Change a $o field to the
                correct value and then Set the PvsCollection with the
                result.

Get-PvsCollection -CollectionName theCollection -SiteName theSite -
                Fields AutoAddSuffix | foreach { $o = $_;
                $o.AutoAddSuffix = "Ex"; $o } | Set-PvsCollection

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y
                and returns the object again so it can be piped to
                the Set command for update.
```

EXAMPLE 3: Get PvsCollection and Enable

Get all PvsCollection that are not Enabled and then Enables them.

```
Get-PvsCollection -Fields Enabled | Where-Object {$_.Enabled -eq
                $false} | foreach { $o = $_; $o.Enabled = $true; $o }
                | Set-PvsCollection
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsConnection

Set the SoapServer connection, and if -Persist is specified the connection settings are saved in the registry. A PvsConnection object can be used as the parameter.

Required

PvsConnection Connection: PvsConnection object with changed property value(s) to be set. The object can come from a pileline.

These values are in the PvsConnection object, and only will be set if the value has changed.

string Name or Server: Name or IP of the Server to connect to. Default=localhost

string Port: The Port to use to connect. Default=54321

string User: User name to use for Authentication. If it has a value, it will be *****. Default=""

string Domain: Domain name to use for Authentication. If it has a value, it will be *****. Default=""

string Password: Password to use for Authentication. If it has a value, it will be *****. Default=""

string Persist: True when the connection settings should be, for Set, or have been, for Get, saved to the registry.

PvsConnection can be created or modified using methods below:

New-Object Citrix.PVS.SnapIn.PvsConnection: Creates default Server=localhost, Port=54321, and no authentication.

New-Object Citrix.PVS.SnapIn.PvsConnection(Citrix.PVS.SnapIn copyFrom): Creates with settings of the copyFrom Citrix.PVS.SnapIn.

SetServerToLocalHostDefaultSettings: Server=localhost, Port=54321, and no authentication.

Copy(Citrix.PVS.SnapIn copyFrom): Modifies the settings to match the copyFrom Citrix.PVS.SnapIn.

Equals(Citrix.PVS.SnapIn compareTo): Returns true when the settings match what is in the compareTo.

When Connection is not passed, the parameters below are used:

Optional field values to set:

string Name or Server: Name or IP of the Server to connect to. Default=localhost

string Port: The Port to use to connect. Default=54321

string User: User name to use for Authentication. If it has a value, it will be *****. Default=""

string Domain: Domain name to use for Authentication. If it has a value, it will be *****. Default=""

string Password: Password to use for Authentication. If it has a value, it will be *****. Default=""

string Persist: True when the connection settings should be, for Set, or have been, for Get, saved to the registry.

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsConnection object is returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsConnection for Individual Fields

Get the PvsConnection into a $o variable. Change the $o field values and then Set the PvsConnection with the result.

```
$o = Get-PvsConnection -Fields Port
$o.Port = 54322
Set-PvsConnection $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsConnection for a Field Using Pipe

Get the PvsConnection into a $o variable for the field that has the wrong value. Change the $o field to the correct value and then Set the PvsConnection with the result.

```
Get-PvsConnection -Fields Port | Where-Object {$_.Port -ne 54322} | foreach { $o = $_; $o.Port = 54322; $o } | Set-PvsConnection
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

EXAMPLE 3: Set PvsConnection Port with Parameter

Set the PvsConnection Port using the Port parameter instead of a PvsConnection object.

Set-PvsConnection -Port 54322

This is the only Set command that has field parameters.

Set-PvsDevice

Set Device(s) changed values from PvsDevice object(s), or set one or more field values for one or more PvsDevices.

Required

PvsDevice Device: PvsDevice object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsDevice object, and only will be set if the value has changed.

string Name or DeviceName: Computer name with no spaces. ASCII computer name characters Max Length=15

string Description: User description. Default="" Max Length=250

PvsPhysicalAddress DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device.

uint BootFrom: Device to boot from. Choices are 1 for vDisk, 2 for Hard Disk, and 3 for Floppy. This cannot be Set for a Device with Personal vDisk. Min=1, Max=3, Default=1

string ClassName: Used by Automatic Update feature to match new versions of Disks to a Device. This cannot be Set for a Device with Personal vDisk. Default="" Max Length=41

uint Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

bool Enabled: True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true

bool LocalDiskEnabled: If there is a local disk menu choice for the Device, this is true. This cannot be Set for a Device with Personal vDisk. Default=false

uint Authentication: Device log in authentication. Choices are 0 for none, 1 for User Name/Password, and 2 for Extern. This cannot be Set for a Device with Personal vDisk. Min=0, Max=2, Default=0

string User: Name of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=20

string Password: Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=100

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

string AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" ASCII Max Length=256

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

string DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

uint Type: 1 when it performs test of Disks, 2 when it performs maintenance on Disks, 3 when it has a Personal vDisk, 4 when it has a Personal vDisk and performs tests, 0 otherwise. A Device with type 0 - 3 can only be Set to 0 - 3, and a Device with type 3 - 4 can only be Set to 3 - 4. Min=0, Max=4, Default=0

uint LocalWriteCacheDiskSize: The size in GB to format the Device cache file disk. If the value is 0, then the disk is not formatted. Min=0, Max=2048, Default=0

bool BdmBoot: Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false

uint BdmType: Use PXE boot when set to 0, BDM (Bios) boot when set to 1 and BDM (Uefi) boot when set to 2. Default=0

uint BdmFormat: 1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0

DateTime BdmUpdated: Timestamp of the last BDM boot disk update. Default=Empty

DateTime BdmCreated: Timstamp when BDM device was created Default=Empty

Guid XsPvsProxyUuid: UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000

string EnableXsProxy: Enable XenServerProxy when set to 1 Default=""

When Device is not passed, the parameters below are used:

One of these required

Guid Guid or DeviceId: GUID of the Device to Set.

string Name or DeviceName: Name of the Device to Set.

PvsPhysicalAddress DeviceMac: MAC of the Device to Set.

Guid CollectionId: GUID of the Collection to set all Devices. DeviceName and DeviceMac cannot be set.

Guid SiteViewId: GUID of the Site View to set all Devices. DeviceName and DeviceMac cannot be set.

Guid FarmViewId: GUID of the Farm View to set all Devices. DeviceName and DeviceMac cannot be set.

string FarmViewName: Name of the Farm View to set all Devices. DeviceName and DeviceMac cannot be set.

or one of these required & resolutions

string CollectionName: Name of the Collection to set all Devices. DeviceName and DeviceMac cannot be set.

string SiteViewName: Name of the Site View to set all Devices. DeviceName and DeviceMac cannot be set.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

Optional field values to set:

string NewName: Computer name with no spaces. ASCII computer name characters Max Length=15

string Description: User description. Default="" Max Length=250

PvsPhysicalAddress DeviceMac: Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device.

uint BootFrom: Device to boot from. Choices are 1 for vDisk, 2 for Hard Disk, and 3 for Floppy. This cannot be Set for a Device with Personal vDisk. Min=1, Max=3, Default=1

string ClassName: Used by Automatic Update feature to match new versions of Disks to a Device. This cannot be Set for a Device with Personal vDisk. Default="" Max Length=41

uint Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

bool Enabled: True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true

bool LocalDiskEnabled: If there is a local disk menu choice for the Device, this is true. This cannot be Set for a Device with Personal vDisk. Default=false

uint Authentication: Device log in authentication. Choices are 0 for none, 1 for User Name/Password, and 2 for Extern. This cannot be Set for a Device with Personal vDisk. Min=0, Max=2, Default=0

string User: Name of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=20

string Password: Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default="" ASCII Max Length=100

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

string AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" ASCII Max Length=256

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

string DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

uint Type: 1 when it performs test of Disks, 2 when it performs maintenance on Disks, 3 when it has a Personal vDisk, 4 when it has a Personal vDisk and performs tests, 0 otherwise. A Device with type 0 - 3 can only be Set to 0 - 3, and a Device with type 3 - 4 can only be Set to 3 - 4. Min=0, Max=4, Default=0

uint LocalWriteCacheDiskSize: The size in GB to format the Device cache file disk. If the value is 0, then the disk is not formatted. Min=0, Max=2048, Default=0

bool BdmBoot: Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false

uint BdmType: Use PXE boot when set to 0, BDM (Bios) boot when set to 1 and BDM (Uefi) boot when set to 2. Default=0

uint BdmFormat: 1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0

DateTime BdmUpdated: Timestamp of the last BDM boot disk update. Default=Empty

```
DateTime BdmCreated: Timstamp when BDM device was
                created  Default=Empty

Guid XsPvsProxyUuid: UUID of XenServer PVS_proxy Default=00000000-
                0000-0000-0000-000000000000

string EnableXsProxy: Enable XenServerProxy when set to 1 Default=""


Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting
                PvsDevice object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "low" to
                have confirmation without the Confirm parameter.
```

EXAMPLE 1: Set PvsDevice for Individual Fields

```
Get the PvsDevice into a $o variable. Change the $o field values and
                then Set the PvsDevice with the result.

$o = Get-PvsDevice -Name theDevice -Fields LocalWriteCacheDiskSize

$o.LocalWriteCacheDiskSize = 1024

Set-PvsDevice $o

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 2: Set PvsDevice for a Field Using Pipe

```
Get the PvsDevice into a $o variable. Change a $o field to the correct
                value and then Set the PvsDevice with the result.

Get-PvsDevice -Name theDevice -Fields LocalWriteCacheDiskSize |
                foreach { $o = $_; $o.LocalWriteCacheDiskSize = 1024;
                $o } | Set-PvsDevice

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y
                and returns the object again so it can be piped to
                the Set command for update.
```

EXAMPLE 3: Get PvsDevice and Enable

```
Get all PvsDevice that are not Enabled and then Enables them.

Get-PvsDevice -Fields Enabled | Where-Object {$_.Enabled -eq $false} |
                foreach { $o = $_; $o.Enabled = $true; $o } | Set-
                PvsDevice

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsDeviceBootstrap

Set Device Bootstrap List(s) changed values from PvsDeviceBootstrap object(s).

Required

PvsDeviceBootstrap[] DeviceBootstrap: Array of PvsDeviceBootstrap objects with changed DeviceBootstrap. The object(s) can come from a pileline.

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsDeviceBootstrap object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

These exist in the DeviceBootstrap array within each PvsDeviceBootstrap.

Each array item is a PvsDeviceBootstrapList object.

They are set using the Add, Insert, Remove, Set, and Reorder methods in the PvsDeviceBootstrap.

string Name or Bootstrap: Name of the bootstrap file. Max Length=259

string MenuText: Text that is displayed in the Boot Menu. If this field has no value, the bootstrap value is used. Default="" ASCII Max Length=64

EXAMPLE 1: Set PvsDeviceBootstrap for Individual Fields

The Get-PvsDeviceBootstrap returns a PvsDeviceBootstrap object with a PvsDeviceBootstrapList array in it called DeviceBootstrap.

The DeviceBootstrap is manipulated using the Add, Insert, Remove, Set and Reorder Methods.

The Set-PvsDeviceBootstrap is called with the final result.

```
$o = Get-PvsDeviceBootstrap -Name theDevice

$o.Add("addName", "addValue")

$o.Insert(0, "insertName", "insertValue")

$o.Remove(1)

$o.Set(2, "setValue")

$o.Reorder(0, 1)

Set-PvsDeviceBootstrap $o
```

Set-PvsDevicePersonality

Set Device Personality List(s) changed values from PvsDevicePersonality object(s).

Required

PvsDevicePersonality[] DevicePersonality: Array of PvsDevicePersonality objects with changed DevicePersonality. The object(s) can come from a pileline.

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsDevicePersonality object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

These exist in the DevicePersonality array within each PvsDevicePersonality.

Each array item is a PvsDevicePersonalityList object.

They are set using the Add, Insert, Remove, Set, and Reorder methods in the PvsDevicePersonality.

string Name: Name of the Device personality item. Max Length=250

string Value: Value for the Device personality item. Max Length=1000

EXAMPLE 1: Set PvsDevicePersonality for Individual Fields

The Get-PvsDevicePersonality returns a PvsDevicePersonality object with a PvsDevicePersonalityList array in it called DevicePersonality.

The DevicePersonality is manipulated using the Add, Insert, Remove, Set and Reorder Methods.

The Set-PvsDevicePersonality is called with the final result.

```
$o = Get-PvsDevicePersonality -Name theDevice

$o.Add("addName", "addValue")

$o.Insert(0, "insertName", "insertValue")

$o.Remove(1)

$o.Set(2, "setValue")

$o.Reorder(0, 1)

Set-PvsDevicePersonality $o
```

Set-PvsDisk

Set Disk(s) changed values from PvsDisk object(s), or set one or more field values for a PvsDisk.

Required

PvsDisk Disk: PvsDisk object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsDisk object, and only will be set if the value has changed.

string Class: Class of the Disk. Max Length=40

string ImageType: Type of this image (software type). Max Length=40

UInt64 WriteCacheSize: RAM cache size (MB). Not 0 when used with Cache in Device RAM, and Cache in Device RAM with Overflow on Hard Disk. A value of 0 will disable the RAM use for Cache in Device RAM with Overflow on Hard Disk. Min=0, Max=131072, Default=0

bool AutoUpdateEnabled: Automatically update this image for matching Devices when set to true. Default false

bool ActivationDateEnabled: Use activation date to activate image when set to true. Default false

bool AdPasswordEnabled: Enable AD password management when set to true.

bool HaEnabled: Enable HA when set to true.

bool PrinterManagementEnabled: Invalid printers will be deleted from the Device when set to true.

uint WriteCacheType: 0 (Private), (other values are standard image) 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), or 9 (Cache in Device RAM with Overflow on Hard Disk). Min=0, Max=9, Default=0

uint LicenseMode: 0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0

DateTime ActiveDate: Date to activate the disk if AutoUpdateEnabled and activationDateEnabled are true. Has the date. Empty when the AutoUpdateEnabled or activationDateEnabled are false.

string LongDescription: Description of the Disk. Max Length=399

string OperatingSystem: Operating System of Disk. Max Length=250

string OsType: Operating System Type of Disk. Max Length=40

string SerialNumber: User defined serial number. Max Length=36

string Date: User defined date. Max Length=40

string Author: User defined author. Max Length=40

string Title: User defined title. Max Length=40

string Company: User defined company. Max Length=40

string InternalName: User defined name. Max Length=63

string OriginalFile: User defined original file. Max Length=127

string HardwareTarget: User defined hardware target. Max Length=127

UInt64 MajorRelease: User defined major release number. Min=0, Max=4294967295, Default=0

UInt64 MinorRelease: User defined minor release number. Min=0, Max=4294967295, Default=0

UInt64 Build: User defined build number. Min=0, Max=4294967295, Default=0

string ClearCacheDisabled: Clear cached secrets disabled.

bool VHDX: If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false

When Disk is not passed, the parameters below are used:

This required

Guid Guid or DiskLocatorId: GUID of the Disk Locator.

or this required & resolution

string Name or DiskLocatorName: Name of the Disk Locator.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

One of these resolutions when needed

Guid StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Optional field values to set:

string Class: Class of the Disk. Max Length=40

string ImageType: Type of this image (software type). Max Length=40

UInt64 WriteCacheSize: RAM cache size (MB). Not 0 when used with Cache in Device RAM, and Cache in Device RAM with Overflow on Hard Disk. A value of 0 will disable the RAM use for Cache in Device RAM with Overflow on Hard Disk. Min=0, Max=131072, Default=0

bool AutoUpdateEnabled: Automatically update this image for matching Devices when set to true. Default false

bool ActivationDateEnabled: Use activation date to activate image when set to true. Default false

bool AdPasswordEnabled: Enable AD password management when set to true.

bool HaEnabled: Enable HA when set to true.

bool PrinterManagementEnabled: Invalid printers will be deleted from the Device when set to true.

uint WriteCacheType: 0 (Private), (other values are standard image) 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), or 9 (Cache in Device RAM with Overflow on Hard Disk). Min=0, Max=9, Default=0

```
uint LicenseMode: 0 (None), 1 (Multiple Activation Key), or 2 (Key
                  Management Service). Min=0, Max=2, Default=0

DateTime ActiveDate: Date to activate the disk if AutoUpdateEnabled
                  and activationDateEnabled are true. Has the date.
                  Empty when the AutoUpdateEnabled or
                  activationDateEnabled are false.

string LongDescription: Description of the Disk. Max Length=399

string OperatingSystem: Operating System of Disk. Max Length=250

string OsType: Operating System Type of Disk. Max Length=40

string SerialNumber: User defined serial number. Max Length=36

string Date: User defined date. Max Length=40

string Author: User defined author. Max Length=40

string Title: User defined title. Max Length=40

string Company: User defined company. Max Length=40

string InternalName: User defined name. Max Length=63

string OriginalFile: User defined original file. Max Length=127

string HardwareTarget: User defined hardware target. Max Length=127

UInt64 MajorRelease: User defined major release number. Min=0,
                  Max=4294967295, Default=0

UInt64 MinorRelease: User defined minor release number. Min=0,
                  Max=4294967295, Default=0

UInt64 Build: User defined build number. Min=0, Max=4294967295,
                  Default=0

string ClearCacheDisabled: Clear cached secrets disabled.

bool VHDX: If VHDX is true, the format of the image is VHDX. Otherwise
                  it is VHD. Default=false


Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting
                  PvsDisk object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -
                  Confirm is specified, the operation will be
                  confirmed. $ConfirmPreference can be set to "low" to
                  have confirmation without the Confirm parameter.
```

EXAMPLE 1: Set PvsDisk for Individual Fields

```
Get the PvsDisk into a $o variable. Change the $o field values and
                  then Set the PvsDisk with the result.

$o = Get-PvsDisk -DiskLocatorId "81ea9077-598d-459f-a443-71fabd1840bf"
                  -Fields PrinterManagementEnabled

$o.PrinterManagementEnabled = $false

Set-PvsDisk $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsDisk for a Field Using Pipe

Get the PvsDisk into a $o variable. Change a $o field to the correct value and then Set the PvsDisk with the result.

```
Get-PvsDisk -DiskLocatorId "81ea9077-598d-459f-a443-71fabd1840bf" -
Fields PrinterManagementEnabled | foreach { $o = $_;
$o.PrinterManagementEnabled = $false; $o } | Set-
PvsDisk
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsDiskLocator

Set Disk Locator(s) changed values from PvsDiskLocator object(s), or set one or more field values for a PvsDiskLocator.

Required

PvsDiskLocator DiskLocator: PvsDiskLocator object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsDiskLocator object, and only will be set if the value has changed.

string Description: User description. Default="" Max Length=250

string MenuText: Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default="" ASCII Max Length=64

Guid ServerId: GUID of the single Server that this Disk Locator is assigned to. It is not used with ServerName. Default=00000000-0000-0000-0000-000000000000

string ServerName: Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=""

bool Enabled: True when this disk can be booted, false otherwise. Default=true

bool RebalanceEnabled: True when this Server can automatically rebalance Devices, false otherwise. Default=false

uint RebalanceTriggerPercent: Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25

uint SubnetAffinity: Qualifier for subnet affinity when assigning a Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2, Default=0

When DiskLocator is not passed, the parameters below are used:

```
This required

 Guid Guid or DiskLocatorId: GUID of the Disk Locator to Set.

or this required & resolution

 string Name or DiskLocatorName: Name of the Disk Locator File to Set.

One of these resolutions when needed

 Guid SiteId: GUID of the Site.

 string SiteName: Name of the Site.

One of these resolutions when needed

 Guid StoreId: GUID of the Store that is needed when a DiskLocatorName
                 is used.

 string StoreName: Name of the Store that is needed when a
                 DiskLocatorName is used.

Optional field values to set:

 string Description: User description. Default="" Max Length=250

 string MenuText: Text that is displayed in the Boot Menu. If this
                 field has no value, the name value is used.
                 Default="" ASCII Max Length=64

 Guid ServerId: GUID of the single Server that this Disk Locator is
                 assigned to. It is not used with ServerName.
                 Default=00000000-0000-0000-0000-000000000000

 string ServerName: Name of the single Server that this Disk Locator is
                 assigned to. It is not used with ServerId. Default=""

 bool Enabled: True when this disk can be booted, false otherwise.
                 Default=true

 bool RebalanceEnabled: True when this Server can automatically
                 rebalance Devices, false otherwise. Default=false

 uint RebalanceTriggerPercent: Percent over fair load that triggers a
                 dynamic Device rebalance. Min=5, Max=5000, Default=25

 uint SubnetAffinity: Qualifier for subnet affinity when assigning a
                 Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2,
                 Default=0


Optional

 SwitchParameter PassThru: If -PassThru is specified, the resulting
                 PvsDiskLocator object(s) are returned.

 SwitchParameter Confirm: The impact of this operation is "low". If -
                 Confirm is specified, the operation will be
                 confirmed. $ConfirmPreference can be set to "low" to
                 have confirmation without the Confirm parameter.
```

EXAMPLE 1: Set PvsDiskLocator for Individual Fields

```
 Get the PvsDiskLocator into a $o variable. Change the $o field values
                 and then Set the PvsDiskLocator with the result.
```

```
$o = Get-PvsDiskLocator -DiskLocatorId "81ea9077-598d-459f-a443-
                71fabd1840bf" -Fields RebalanceEnabled

$o.RebalanceEnabled = $true

Set-PvsDiskLocator $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsDiskLocator for a Field Using Pipe

Get the PvsDiskLocator into a $o variable. Change a $o field to the correct value and then Set the PvsDiskLocator with the result.

```
Get-PvsDiskLocator -DiskLocatorId "81ea9077-598d-459f-a443-
                71fabd1840bf" -Fields RebalanceEnabled | foreach { $o
                = $_; $o.RebalanceEnabled = $true; $o } | Set-
                PvsDiskLocator
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

EXAMPLE 3: Get PvsDiskLocator and Enable

Get all PvsDiskLocator that are not Enabled and then Enables them.

```
Get-PvsDiskLocator -Fields Enabled | Where-Object {$_.Enabled -eq
                $false} | foreach { $o = $_; $o.Enabled = $true; $o }
                | Set-PvsDiskLocator
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsDiskUpdateDevice

Set Disk Update Device(s) changed values from PvsDiskUpdateDevice object(s), or set one or more field values for one or more PvsDiskUpdateDevices.

Required

PvsDiskUpdateDevice DiskUpdateDevice: PvsDiskUpdateDevice object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsDiskUpdateDevice object, and only will be set if the value has changed.

string Description: User description. Default="" Max Length=250

uint Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

string DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

string AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" Max Length=256

When DiskUpdateDevice is not passed, the parameters below are used:

One of these required

Guid Guid or DeviceId: GUID of the Disk Update Device to Set.

string Name or DeviceName: Name of Disk Update Device to Set.

PvsPhysicalAddress DeviceMac: MAC of Disk Update Device to Set.

Guid SiteId: GUID of the Site. Can be used alone to Set all Disk Update Devices in the Site.

string SiteName: Name of the Site. Can be used alone to Set all Disk Update Devices in the Site.

Guid DiskLocatorId: GUID of the DiskLocator to Set the Disk Update Device for.

or this required & resolution

string DiskLocatorName: Name of the DiskLocator to Set the Disk Update Device for.

One of these resolutions when needed

Guid SiteId: GUID of the Site. Can be used alone to Set all Disk Update Devices in the Site.

string SiteName: Name of the Site. Can be used alone to Set all Disk Update Devices in the Site.

One of these resolutions when needed

Guid StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Optional field values to set:

string Description: User description. Default="" Max Length=250

uint Port: UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

uint AdTimestamp: The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

uint AdSignature: The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=0

string DomainName: Fully qualified name of the domain that the Device belongs to. Do not set this field, it is only set internally by PVS. Default="" Max Length=255

string DomainObjectSID: The value of the objectSID AD attribute of the same name for the Device's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=186

string DomainControllerName: The name of the DC used to create the host's computer account. Do not set this field, it is only set internally by PVS. Default="" Max Length=4000

DateTime DomainTimeCreated: The time that the computer account was created. Has the date and time including milliseconds. Do not set this field, it is only set internally by PVS. Default=Empty

string AdPassword: The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default="" Max Length=256

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsDiskUpdateDevice object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsDiskUpdateDevice for Individual Fields

```
Get the PvsDiskUpdateDevice into a $o variable. Change the $o field
                values and then Set the PvsDiskUpdateDevice with the
                result.

$o = Get-PvsDiskUpdateDevice -Name theDevice -Fields Port

$o.Port = 5901

Set-PvsDiskUpdateDevice $o

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 2: Set PvsDiskUpdateDevice for a Field Using Pipe

```
Get the PvsDiskUpdateDevice into a $o variable. Change a $o field to
                the correct value and then Set the
                PvsDiskUpdateDevice with the result.

Get-PvsDiskUpdateDevice -Name theDevice -Fields Port | foreach { $o =
                $_; $o.Port = 5901; $o } | Set-PvsDiskUpdateDevice

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y
                and returns the object again so it can be piped to
                the Set command for update.
```

Set-PvsDiskVersion

```
Set Disk Version(s) changed values from PvsDiskVersion object(s), or set
  one or more field values for a PvsDiskVersion.

   Required

    PvsDiskVersion DiskVersion: PvsDiskVersion object(s) with changed
                property value(s) to be set. The object(s) can come
                from a pileline.

    These values are in the PvsDiskVersion object, and only will be set if
                the value has changed.

    string Description: User description. Default="" Max Length=250

    DateTime ScheduledDate: Date/Time that the Disk Version is scheduled
                to become available. Has the date, hour and minute.
                Empty when the disk version is made available
                immediately. Default=Empty


   When DiskVersion is not passed, the parameters below are used:

   This required

    Guid Guid or DiskLocatorId: GUID of the Disk Locator Version to Set.

   or this required & resolution

    string Name or DiskLocatorName: Name of the Disk Locator Version to
                Set.
```

```
This required

 uint Version: Version to Set.

One of these resolutions when needed

 Guid SiteId: GUID of the Site.

 string SiteName: Name of the Site.

One of these resolutions when needed

 Guid StoreId: GUID of the Store that is needed when a DiskLocatorName
                is used.

 string StoreName: Name of the Store that is needed when a
                DiskLocatorName is used.

Optional field values to set:

 string Description: User description. Default="" Max Length=250

 DateTime ScheduledDate: Date/Time that the Disk Version is scheduled
                to become available. Has the date, hour and minute.
                Empty when the disk version is made available
                immediately. Default=Empty


Optional

 SwitchParameter PassThru: If -PassThru is specified, the resulting
                PvsDiskVersion object(s) are returned.

 SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "low" to
                have confirmation without the Confirm parameter.
```

EXAMPLE 1: Set PvsDiskVersion for Individual Fields

```
    Get the PvsDiskVersion into a $o variable. Change the $o field values
                and then Set the PvsDiskVersion with the result.

    $o = Get-PvsDiskVersion -DiskLocatorId "81ea9077-598d-459f-a443-
                71fabd1840bf" -Version 5 -Fields ScheduledDate

    $o.ScheduledDate = "2016/01/01"

    Set-PvsDiskVersion $o

    The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 2: Set PvsDiskVersion for a Field Using Pipe

```
    Get the PvsDiskVersion into a $o variable. Change a $o field to the
                correct value and then Set the PvsDiskVersion with
                the result.

    Get-PvsDiskVersion -DiskLocatorId "81ea9077-598d-459f-a443-
                71fabd1840bf" -Version 5 -Fields ScheduledDate |
                foreach { $o = $_; $o.ScheduledDate = "2016/01/01";
                $o } | Set-PvsDiskVersion
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsFarm

Set Farm changed values from a PvsFarm object, or set one or more field values for a PvsFarm.

Required

PvsFarm Farm: PvsFarm object with changed property value(s) to be set. The object can come from a pileline.

These values are in the PvsFarm object, and only will be set if the value has changed.

string Name or FarmName: Name of the Farm. Default="" Max Length=50

string Description: User description. Default="" Max Length=250

bool AutoAddEnabled: True when Auto Add is enabled, false otherwise. Default=false

bool AuditingEnabled: True when Auditing is enabled, false otherwise. Default=false

DateTime LastAuditArchiveDate: Last date of Audit Trail data that was Archived. Has the date. Default=Empty

Guid DefaultSiteId: GUID of the Site to place new Devices into automatically. Not used with defaultSiteName. Default=00000000-0000-0000-0000-000000000000

string DefaultSiteName: Name of the Site to place new Devices into automatically. Not used with DefaultSiteId. Default=""

bool OfflineDatabaseSupportEnabled: True when Offline Database Support is enabled, false otherwise. Default=false

string LicenseServer: License server name. Default="" Max Length=255

uint LicenseServerPort: License server port. Min=1025, Max=65534, Default=27000

bool LicenseTradeUp: License server trade up, when set to true. Default=false

bool AutomaticMergeEnabled: True when Automatic Merge is enabled, false otherwise. If the number of versions becomes more than the MaxVersions value, a merge will occur at the end of PromoteDiskVersion. Default=true

uint MaxVersions: Maximum number a versions of a Disk that can exist before a merge will automatically occur. Min=3, Max=50, Default=5

uint MergeMode: Mode to place the version in after a merge has occurred. Values are: 0 (Production), 1 (Test) and 2 (Maintenance). Min=0, Max=2, Default=2

When Farm is not passed, the parameters below are used:

Optional

Guid Guid or FarmId: GUID of the Farm to Set. This is optional since there is only one Farm.

Optional field values to set:

string NewName: Name of the Farm. Default="" Max Length=50

string Description: User description. Default="" Max Length=250

bool AutoAddEnabled: True when Auto Add is enabled, false otherwise. Default=false

bool AuditingEnabled: True when Auditing is enabled, false otherwise. Default=false

DateTime LastAuditArchiveDate: Last date of Audit Trail data that was Archived. Has the date. Default=Empty

Guid DefaultSiteId: GUID of the Site to place new Devices into automatically. Not used with defaultSiteName. Default=00000000-0000-0000-0000-000000000000

string DefaultSiteName: Name of the Site to place new Devices into automatically. Not used with DefaultSiteId. Default=""

bool OfflineDatabaseSupportEnabled: True when Offline Database Support is enabled, false otherwise. Default=false

string LicenseServer: License server name. Default="" Max Length=255

uint LicenseServerPort: License server port. Min=1025, Max=65534, Default=27000

bool LicenseTradeUp: License server trade up, when set to true. Default=false

bool AutomaticMergeEnabled: True when Automatic Merge is enabled, false otherwise. If the number of versions becomes more than the MaxVersions value, a merge will occur at the end of PromoteDiskVersion. Default=true

uint MaxVersions: Maximum number a versions of a Disk that can exist before a merge will automatically occur. Min=3, Max=50, Default=5

uint MergeMode: Mode to place the version in after a merge has occurred. Values are: 0 (Production), 1 (Test) and 2 (Maintenance). Min=0, Max=2, Default=2

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsFarm object is returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsFarm for Individual Fields

```
Get the PvsFarm into a $o variable. Change the $o field values and
                then Set the PvsFarm with the result.

$o = Get-PvsFarm -Fields AuditingEnabled,
                OfflineDatabaseSupportEnabled

$o.AuditingEnabled = $true

$o.OfflineDatabaseSupportEnabled = $true

Set-PvsFarm $o

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 2: Set PvsFarm for a Field Using Pipe

```
Get the PvsFarm into a $o variable for the field that has the wrong
                value. Change the $o field to the correct value and
                then Set the PvsFarm with the result.

Get-PvsFarm -Fields AuditingEnabled | Where-Object {$_.AuditingEnabled
                -ne $true} | foreach { $o = $_; $o.AuditingEnabled =
                $true; $o } | Set-PvsFarm

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y
                and returns the object again so it can be piped to
                the Set command for update.
```

Set-PvsFarmView

```
Set Farm View(s) changed values from PvsFarmView object(s), or set one or
  more field values for a PvsFarmView.

   Required

    PvsFarmView FarmView: PvsFarmView object(s) with changed property
                value(s) to be set. The object(s) can come from a
                pileline.

    These values are in the PvsFarmView object, and only will be set if
                the value has changed.

    string Name or FarmViewName: name of the Farm View. Max Length=50

    string Description: User description. Default="" Max Length=250


   When FarmView is not passed, the parameters below are used:

   One of these required

    Guid Guid or FarmViewId: GUID of the Farm View to Set.

    string Name or FarmViewName: Name of the Farm View to Set.

   Optional field values to set:

    string NewName: name of the Farm View. Max Length=50
```

string Description: User description. Default="" Max Length=250

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsFarmView object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsFarmView for Individual Fields

Get the PvsFarmView into a $o variable. Change the $o field values and then Set the PvsFarmView with the result.

```
$o = Get-PvsFarmView -Name oldFarmView -Fields Name
$o.Name = "newFarmView"
Set-PvsFarmView $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsFarmView for a Field Using Pipe

Get the PvsFarmView into a $o variable. Change a $o field to the correct value and then Set the PvsFarmView with the result.

```
Get-PvsFarmView -Name oldFarmView -Fields Name | foreach { $o = $_; $o.Name = "newFarmView"; $o } | Set-PvsFarmView
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsOverrideVersion

Specify a Disk Version all Production Devices will boot from.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator File to Override the Production Version for.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator File to Override the Production Version for.

Optional

uint Version: Version to set as the Override Production Version. If Version is not included and if there is an Override Production Version, then no longer have it as the Override Version.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set-PvsOverrideVersion for Name with Version

```
Set-PvsOverrideVersion -Name theDiskLocator -SiteName theSite -
StoreName theStore -Version 4
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 2: Set-PvsOverrideVersion for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Set-PvsOverrideVersion.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Fields Guid | Set-PvsOverrideVersion -
Version 4
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 3: Set-PvsOverrideVersion for Name

```
Set-PvsOverrideVersion -Name theDiskLocator -SiteName theSite -
StoreName theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 4: Set-PvsOverrideVersion for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Set-PvsOverrideVersion.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Fields Guid | Set-PvsOverrideVersion
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Set-PvsServer

Set Server(s) changed values from PvsServer object(s), or set one or more field values for a PvsServer. Restart the Server(s) after setting.

Required

PvsServer Server: PvsServer object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsServer object, and only will be set if the value has changed.

string Name or ServerName: Computer name with no spaces. ASCII computer name characters Max Length=21

string Description: User description. Default="" Max Length=250

uint AdMaxPasswordAge: Number of days before a password expires. Min=1, Max=30, Default=7

uint LicenseTimeout: Amount of seconds before a license times out. Min=15, Max=300, Default=30

uint VDiskCreatePacing: VDisk create time pacing in miliseconds. Min=0, Max=5, Default=0

uint FirstPort: Number of the first UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6910

uint LastPort: Number of the last UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6930

uint ThreadsPerPort: Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8

uint BuffersPerThread: Number of buffers per worker thread. Min=1, Max=128, Default=24

uint ServerCacheTimeout: Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8

uint IoBurstSize: Number of bytes read/writes can send in a burst of packets. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=4096, Max=61440, Default=32768

uint MaxTransmissionUnits: Ethernet maximum transmission unit size for the protocol for use for Server and Device. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=502, Max=16426, Default=1506

uint MaxBootDevicesAllowed: Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500

uint MaxBootSeconds: Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60

uint BootPauseSeconds: Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10

bool AdMaxPasswordAgeEnabled: Age the password, when set to true. Default=false

bool EventLoggingEnabled: Enable event logging, when set to true. Default=false

bool NonBlockingIoEnabled: Use non-Blocking IO, when set to true. Default=true

string[] Ip: One or more streaming ip addresses.

uint InitialQueryConnectionPoolSize: Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50

uint InitialTransactionConnectionPoolSize: Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50

uint MaxQueryConnectionPoolSize: Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000

uint MaxTransactionConnectionPoolSize: Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000

uint RefreshInterval: Interval, in number of seconds, the server should wait before refreshing settings. If set to 0, unused database connections are never released. Min=0, Max=32767, Default=300

uint UnusedDbConnectionTimeout: Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300

uint BusyDbConnectionRetryCount: Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2

uint BusyDbConnectionRetryInterval: Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25

uint LocalConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are local. A value of 0 disables the feature. Min=0, Max=128, Default=4

uint RemoteConcurrentIoLimit: Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4

uint LogLevel: Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug), and 6 (Trace). Min=0, Max=6, Default=4

uint LogFileSizeMax: Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5

uint LogFileBackupCopiesMax: Maximum number of log file backups. Min=1, Max=50, Default=4

float PowerRating: A strictly relative rating of this Server's capabilities when compared to other Servers in the Store(s) it belongs too; can be used to help tune load balancing. Min=0.1, Max=1000, Default=1

DateTime LastCeipUploadAttempt: Time that this server last attempted a CEIP upload. Default=Empty

DateTime LastBugReportAttempt: Time that this server last attempted to upload or generate a bug report bundle. Default=Empty

string LastBugReportStatus: Status of the last bug report on this server. Default="" Max Length=250

string LastBugReportResult: Status of the last bug report on this server. Default="" Max Length=4000

string LastBugReportSummary: Summary of the last bug report on this server. Default="" Max Length=250

When Server is not passed, the parameters below are used:

One of these required

Guid Guid or ServerId: GUID of the Server to Set.

string Name or ServerName: Name of the Server to Set.

Optional field values to set:

string NewName: Computer name with no spaces. ASCII computer name characters Max Length=21

string Description: User description. Default="" Max Length=250

uint AdMaxPasswordAge: Number of days before a password expires. Min=1, Max=30, Default=7

uint LicenseTimeout: Amount of seconds before a license times out. Min=15, Max=300, Default=30

uint VDiskCreatePacing: VDisk create time pacing in miliseconds. Min=0, Max=5, Default=0

uint FirstPort: Number of the first UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6910

uint LastPort: Number of the last UDP port for use by the Stream Service, First and Last must allow at least 5 ports. Min=1025, Max=65534, Default=6930

uint ThreadsPerPort: Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8

uint BuffersPerThread: Number of buffers per worker thread. Min=1, Max=128, Default=24

uint ServerCacheTimeout: Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8

uint IoBurstSize: Number of bytes read/writes can send in a burst of packets. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=4096, Max=61440, Default=32768

uint MaxTransmissionUnits: Ethernet maximum transmission unit size for the protocol for use for Server and Device. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=502, Max=16426, Default=1506

uint MaxBootDevicesAllowed: Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500

uint MaxBootSeconds: Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60

uint BootPauseSeconds: Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10

bool AdMaxPasswordAgeEnabled: Age the password, when set to true. Default=false

bool EventLoggingEnabled: Enable event logging, when set to true. Default=false

bool NonBlockingIoEnabled: Use non-Blocking IO, when set to true. Default=true

string[] Ip: One or more streaming ip addresses.

uint InitialQueryConnectionPoolSize: Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50

uint InitialTransactionConnectionPoolSize: Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50

uint MaxQueryConnectionPoolSize: Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000

uint MaxTransactionConnectionPoolSize: Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000

uint RefreshInterval: Interval, in number of seconds, the server should wait before refreshing settings. If set to 0, unused database connections are never released. Min=0, Max=32767, Default=300

uint UnusedDbConnectionTimeout: Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300

uint BusyDbConnectionRetryCount: Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2

uint BusyDbConnectionRetryInterval: Interval, in number of milliseconds, the server should wait before retrying

```
                to connect to a database. Min=0, Max=10000,
                Default=25

    uint LocalConcurrentIoLimit: Maximum concurrent IO transactions it
                performs for vDisks that are local. A value of 0
                disables the feature. Min=0, Max=128, Default=4

    uint RemoteConcurrentIoLimit: Maximum concurrent IO transactions it
                performs for vDisks that are remote. A value of 0
                disables the feature. Min=0, Max=128, Default=4

    uint LogLevel: Level to perform logging at. Values are: 0 (None), 1
                (Fatal), 2 (Error), 3 (Warning), 4 (Info), 5 (Debug),
                and 6 (Trace). Min=0, Max=6, Default=4

    uint LogFileSizeMax: Maximum size log files can reach in Megabytes.
                Min=1, Max=50, Default=5

    uint LogFileBackupCopiesMax: Maximum number of log file backups.
                Min=1, Max=50, Default=4

    float PowerRating: A strictly relative rating of this Server's
                capabilities when compared to other Servers in the
                Store(s) it belongs too; can be used to help tune
                load balancing. Min=0.1, Max=1000, Default=1

    DateTime LastCeipUploadAttempt: Time that this server last attempted a
                CEIP upload. Default=Empty

    DateTime LastBugReportAttempt: Time that this server last attempted to
                upload or generate a bug report bundle. Default=Empty

    string LastBugReportStatus: Status of the last bug report on this
                server. Default="" Max Length=250

    string LastBugReportResult: Status of the last bug report on this
                server. Default="" Max Length=4000

    string LastBugReportSummary: Summary of the last bug report on this
                server. Default="" Max Length=250


Optional

    SwitchParameter PassThru: If -PassThru is specified, the resulting
                PvsServer object(s) are returned.

    SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "low" to
                have confirmation without the Confirm parameter.
```

EXAMPLE 1: Set PvsServer for Individual Fields

```
    Get the PvsServer into a $o variable. Change the $o field values and
                then Set the PvsServer with the result.

    $o = Get-PvsServer -Name theServer -Fields LicenseTimeout,
                ThreadsPerPort

    $o.LicenseTimeout = 60

    $o.ThreadsPerPort = 10

    Set-PvsServer $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsServer for a Field Using Pipe

Get the PvsServer into a $o variable. Change a $o field to the correct value and then Set the PvsServer with the result.

```
Get-PvsServer -Name theServer -Fields LicenseTimeout | foreach { $o = $_; $o.LicenseTimeout = 60; $o } | Set-PvsServer
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsServerBiosBootstrap

Oem Only: Set Server Bios Bootstrap(s) changed values from PvsServerBiosBootstrap object(s), or set one or more field values for a PvsServerBiosBootstrap.

Required

PvsServerBiosBootstrap ServerBiosBootstrap: PvsServerBiosBootstrap object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsServerBiosBootstrap object, and only will be set if the value has changed.

bool Enabled: Automatically update the BIOS on the target device with these setting when set to true, otherwise do not use these settings. Default=false

bool DhcpEnabled: Use DHCP to retrieve target device IP when set to true, otherwise use the static domain, dnsIpAddresstrue and dnsIpAddress2 settings. Default=true

bool Lookup: Use DNS to find the Server when set to true with the ServerName host value, otherwise use the bootservertrue_Ip, bootservertrue_Port, bootserver2_Ip, bootserver2_Port, bootserver3_Ip, bootserver3_Port, bootserver4_Ip, and bootserver4_Port settings. Default=true

bool VerboseMode: Display verbose diagnostic information when set to true. Default=false

bool InterruptSafeMode: Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false

bool PaeMode: PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false

bool BootFromHdOnFail: For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false

uint RecoveryTime: When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50

uint PollingTimeout: Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000

uint GeneralTimeout: Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000

string Name or ServerName: Host to use for DNS lookup. Only used when Lookup is true. Default=IMAGESERVER1

string Bootserver1_Ip: 1st boot server IP. Only used when Lookup is false.

uint Bootserver1_Port: 1st boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

string Bootserver2_Ip: 2nd boot server IP. Only used when Lookup is false. Default=0.0.0.0

uint Bootserver2_Port: 2nd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

string Bootserver3_Ip: 3rd boot server IP. Only used when Lookup is false. Default=0.0.0.0

uint Bootserver3_Port: 3rd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

string Bootserver4_Ip: 4th boot server IP. Only used when Lookup is false. Default=0.0.0.0

uint Bootserver4_Port: 4th boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

string Domain: Domain of the primary and secondary DNS servers. Only used when DhcpEnabled is false.

string DnsIpAddress1: Primary DNS server IP. Only used when DhcpEnabled is false.

string DnsIpAddress2: Secondary DNS server IP. Only used when DhcpEnabled is false.

When ServerBiosBootstrap is not passed, the parameters below are used:

One of these required

Guid Guid or ServerId: GUID of the Server to Set the dell_bios.bin BIOS bootstrap file.

string Name or ServerName: Name of the Server to Set the dell_bios.bin BIOS bootstrap file.

Optional field values to set:

bool Enabled: Automatically update the BIOS on the target device with these setting when set to true, otherwise do not use these settings. Default=false

bool DhcpEnabled: Use DHCP to retrieve target device IP when set to true, otherwise use the static domain,

dnsIpAddresstrue and dnsIpAddress2 settings. Default=true

bool Lookup: Use DNS to find the Server when set to true with the ServerName host value, otherwise use the bootservertrue_Ip, bootservertrue_Port, bootserver2_Ip, bootserver2_Port, bootserver3_Ip, bootserver3_Port, bootserver4_Ip, and bootserver4_Port settings. Default=true

bool VerboseMode: Display verbose diagnostic information when set to true. Default=false

bool InterruptSafeMode: Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false

bool PaeMode: PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false

bool BootFromHdOnFail: For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false

uint RecoveryTime: When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50

uint PollingTimeout: Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000

uint GeneralTimeout: Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000

string NewName: Host to use for DNS lookup. Only used when Lookup is true. Default=IMAGESERVER1

string Bootserver1_Ip: 1st boot server IP. Only used when Lookup is false.

uint Bootserver1_Port: 1st boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

string Bootserver2_Ip: 2nd boot server IP. Only used when Lookup is false. Default=0.0.0.0

uint Bootserver2_Port: 2nd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

string Bootserver3_Ip: 3rd boot server IP. Only used when Lookup is false. Default=0.0.0.0

uint Bootserver3_Port: 3rd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

string Bootserver4_Ip: 4th boot server IP. Only used when Lookup is false. Default=0.0.0.0

uint Bootserver4_Port: 4th boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

string Domain: Domain of the primary and secondary DNS servers. Only used when DhcpEnabled is false.

string DnsIpAddress1: Primary DNS server IP. Only used when DhcpEnabled is false.

string DnsIpAddress2: Secondary DNS server IP. Only used when DhcpEnabled is false.

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsServerBiosBootstrap object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsServerBiosBootstrap for Individual Fields

Get the PvsServerBiosBootstrap into a $o variable. Change the $o field values and then Set the PvsServerBiosBootstrap with the result.

```
$o = Get-PvsServerBiosBootstrap -Name theServer -Fields DhcpEnabled

$o.DhcpEnabled = $true

Set-PvsServerBiosBootstrap $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsServerBiosBootstrap for a Field Using Pipe

Get the PvsServerBiosBootstrap into a $o variable. Change a $o field to the correct value and then Set the PvsServerBiosBootstrap with the result.

```
Get-PvsServerBiosBootstrap -Name theServer -Fields DhcpEnabled |
foreach { $o = $_; $o.DhcpEnabled = $true; $o } |
Set-PvsServerBiosBootstrap
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsServerBootstrap

Set Server Bootstrap(s) changed values from PvsServerBootstrap object(s), or set one or more field values for a PvsServerBootstarp.

Required

PvsServerBootstrap ServerBootstrap: PvsServerBootstrap object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsServerBootstrap object, and only will be set if the value has changed.

string Bootserver1_Ip: 1st boot server IP.

string Bootserver1_Netmask: 1st boot server netmask. Default=0.0.0.0

string Bootserver1_Gateway: 1st boot server gateway. Default=0.0.0.0

uint Bootserver1_Port: 1st boot server port. Min=1025, Max=65536, Default=6910

string Bootserver2_Ip: 2nd boot server IP. Default=0.0.0.0

string Bootserver2_Netmask: 2nd boot server netmask. Default=0.0.0.0

string Bootserver2_Gateway: 2nd boot server gateway. Default=0.0.0.0

uint Bootserver2_Port: 2nd boot server port. Min=1025, Max=65536, Default=6910

string Bootserver3_Ip: 3rd boot server IP. Default=0.0.0.0

string Bootserver3_Netmask: 3rd boot server netmask. Default=0.0.0.0

string Bootserver3_Gateway: 3rd boot server gateway. Default=0.0.0.0

uint Bootserver3_Port: 3rd boot server port. Min=1025, Max=65536, Default=6910

string Bootserver4_Ip: 4th boot server IP. Default=0.0.0.0

string Bootserver4_Netmask: 4th boot server netmask. Default=0.0.0.0

string Bootserver4_Gateway: 4th boot server gateway. Default=0.0.0.0

uint Bootserver4_Port: 4th boot server port. Min=1025, Max=65536, Default=6910

bool VerboseMode: Display verbose diagnostic information when set to true. Default=false

bool InterruptSafeMode: Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false

bool PaeMode: PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false

bool BootFromHdOnFail: For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false

uint RecoveryTime: When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50

uint PollingTimeout: Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000

uint GeneralTimeout: Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000

When ServerBootstrap is not passed, the parameters below are used:

One of these required

Guid Guid or ServerId: GUID of the Server to Set the named bootstrap file on.

string ServerName: Name of the Server to Set the named bootstrap file on.

This required

string Name: Name of the bootstrap file.

Optional field values to set:

string Bootserver1_Ip: 1st boot server IP.

string Bootserver1_Netmask: 1st boot server netmask. Default=0.0.0.0

string Bootserver1_Gateway: 1st boot server gateway. Default=0.0.0.0

uint Bootserver1_Port: 1st boot server port. Min=1025, Max=65536, Default=6910

string Bootserver2_Ip: 2nd boot server IP. Default=0.0.0.0

string Bootserver2_Netmask: 2nd boot server netmask. Default=0.0.0.0

string Bootserver2_Gateway: 2nd boot server gateway. Default=0.0.0.0

uint Bootserver2_Port: 2nd boot server port. Min=1025, Max=65536, Default=6910

string Bootserver3_Ip: 3rd boot server IP. Default=0.0.0.0

string Bootserver3_Netmask: 3rd boot server netmask. Default=0.0.0.0

string Bootserver3_Gateway: 3rd boot server gateway. Default=0.0.0.0

uint Bootserver3_Port: 3rd boot server port. Min=1025, Max=65536, Default=6910

string Bootserver4_Ip: 4th boot server IP. Default=0.0.0.0

string Bootserver4_Netmask: 4th boot server netmask. Default=0.0.0.0

string Bootserver4_Gateway: 4th boot server gateway. Default=0.0.0.0

uint Bootserver4_Port: 4th boot server port. Min=1025, Max=65536, Default=6910

bool VerboseMode: Display verbose diagnostic information when set to true. Default=false

bool InterruptSafeMode: Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false

bool PaeMode: PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false

bool BootFromHdOnFail: For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false

uint RecoveryTime: When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50

uint PollingTimeout: Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000

uint GeneralTimeout: Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsServerBootstrap object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsServerBootstrap for Individual Fields

Get the PvsServerBootstrap into a $o variable. Change the $o field values and then Set the PvsServerBootstrap with the result.

```
$o = Get-PvsServerBootstrap -Name theBootstrapFile -ServerName theServer -Fields VerboseMode

$o.VerboseMode = $true

Set-PvsServerBootstrap $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsServerBootstrap for a Field Using Pipe

Get the PvsServerBootstrap into a $o variable. Change a $o field to the correct value and then Set the PvsServerBootstrap with the result.

```
Get-PvsServerBootstrap -Name theBootstrapFile -ServerName theServer -Fields VerboseMode | foreach { $o = $_; $o.VerboseMode = $true; $o } | Set-PvsServerBootstrap
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsServerStore

Set ServerStore(s) changed values from PvsServerStore object(s), or set one or more field values for a PvsServerStore.

Required

PvsServerStore ServerStore: PvsServerStore object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsServerStore object, and only will be set if the value has changed.

string Path: Directory path that the Server uses to access the Store. Default="" Max Length=255

string[] CachePath: Cache path(s) that the Server uses with the Store. If none are specified the caches will be placed in the Store cachePath. Default=None

When ServerStore is not passed, the parameters below are used:

One of these required

Guid Guid or ServerId: GUID of a Server that uses the path to get to the Store.

string Name or ServerName: Name of a Server that uses the path to get to the Store.

One of these required

Guid StoreId: GUID of the Store.

string StoreName: Name of the Store.

Optional field values to set:

string Path: Directory path that the Server uses to access the Store. Default="" Max Length=255

string[] CachePath: Cache path(s) that the Server uses with the Store. If none are specified the caches will be placed in the Store cachePath. Default=None

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsServerStore object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsServerStore for Individual Fields

Get the PvsServerStore into a $o variable. Change the $o field values and then Set the PvsServerStore with the result.

```
$o = Get-PvsServerStore -ServerName theServer -StoreName theStore -Fields CachePath

$o.CachePath = "\\Network\CachePath"

Set-PvsServerStore $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsServerStore for a Field Using Pipe

Get the PvsServerStore into a $o variable. Change a $o field to the correct value and then Set the PvsServerStore with the result.

```
Get-PvsServerStore -ServerName theServer -StoreName theStore -Fields CachePath | foreach { $o = $_; $o.CachePath = "\\Network\CachePath"; $o } | Set-PvsServerStore
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsSite

Set Site(s) changed values from PvsSite object(s), or set one or more field values for a PvsSite.

Required

PvsSite Site: PvsSite object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsSite object, and only will be set if the value has changed.

string Name or SiteName: Name of the Site. Max Length=50

string Description: User description. Default="" Max Length=250

Guid DefaultCollectionId: GUID of the Collection to place new Devices into automatically. Not used with defaultCollectionName. Default=00000000-0000-0000-0000-000000000000

string DefaultCollectionName: Name of the Collection to place new Devices into automatically. Not used with DefaultCollectionId. Default=""

uint InventoryFilePollingInterval: The number of seconds between polls for Disk changes in the Stores. Min=1, Max=600, Default=60

bool EnableDiskUpdate: True when Disk Updated is enabled for the Site, false otherwise. Default=false

Guid DiskUpdateServerId: GUID of the Disk Update Server for the Site. Not used with DiskUpdateServerName. Default=00000000-0000-0000-0000-000000000000

string DiskUpdateServerName: Name of the Disk Update Server for the Site. Not used with DiskUpdateServerId. Default=""

string MakUser: User name used for MAK activation. Default="" Max Length=64

string MakPassword: User password used for MAK activation. Default="" Max Length=64

string EnableXsProxy: Enable XenServerProxy when set to 1 Default=""

Guid VirtualHostingPoolId: GUID of the VirtualHostingPool object.

string VirtualHostingPoolName: Name of the VirtualHostingPool object.

string XsPvsSiteUuid: GUID of the XenServer PVS Site.

When Site is not passed, the parameters below are used:

One of these required

Guid Guid or SiteId: GUID of the Site to Set.

string Name or SiteName: Name of the Site to Set.

Optional field values to set:

string NewName: Name of the Site. Max Length=50

string Description: User description. Default="" Max Length=250

Guid DefaultCollectionId: GUID of the Collection to place new Devices into automatically. Not used with defaultCollectionName. Default=00000000-0000-0000-0000-000000000000

string DefaultCollectionName: Name of the Collection to place new Devices into automatically. Not used with DefaultCollectionId. Default=""

uint InventoryFilePollingInterval: The number of seconds between polls for Disk changes in the Stores. Min=1, Max=600, Default=60

bool EnableDiskUpdate: True when Disk Updated is enabled for the Site, false otherwise. Default=false

Guid DiskUpdateServerId: GUID of the Disk Update Server for the Site. Not used with DiskUpdateServerName. Default=00000000-0000-0000-0000-000000000000

string DiskUpdateServerName: Name of the Disk Update Server for the Site. Not used with DiskUpdateServerId. Default=""

string MakUser: User name used for MAK activation. Default="" Max Length=64

string MakPassword: User password used for MAK activation. Default="" Max Length=64

string EnableXsProxy: Enable XenServerProxy when set to 1 Default=""

Guid VirtualHostingPoolId: GUID of the VirtualHostingPool object.

string VirtualHostingPoolName: Name of the VirtualHostingPool object.

string XsPvsSiteUuid: GUID of the XenServer PVS Site.

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsSite object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsSite for Individual Fields

Get the PvsSite into a $o variable. Change the $o field values and then Set the PvsSite with the result.

```
$o = Get-PvsSite -Name theSite -Fields InventoryFilePollingInterval
```

```
$o.InventoryFilePollingInterval = 120

Set-PvsSite $o

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 2: Set PvsSite for a Field Using Pipe

```
Get the PvsSite into a $o variable. Change a $o field to the correct
                value and then Set the PvsSite with the result.

Get-PvsSite -Name theSite -Fields InventoryFilePollingInterval |
                foreach { $o = $_; $o.InventoryFilePollingInterval =
                120; $o } | Set-PvsSite

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y
                and returns the object again so it can be piped to
                the Set command for update.
```

Set-PvsSiteView

```
Set Site View(s) changed values from PvsSiteView object(s), or set one or
  more field values for a PvsSiteView.

  Required

   PvsSiteView SiteView: PvsSiteView object(s) with changed property
                value(s) to be set. The object(s) can come from a
                pileline.

   These values are in the PvsSiteView object, and only will be set if
                the value has changed.

   string Name or SiteViewName: Name of the Site View. Max Length=50

   string Description: User description. Default="" Max Length=250


  When SiteView is not passed, the parameters below are used:

  This required

   Guid Guid or SiteViewId: GUID of the Site View to Set.

  or this required & resolution

   string Name or SiteViewName: Name of the Site View to Set.

  One of these resolutions when needed

   Guid SiteId: GUID of the Site.

   string SiteName: Name of the Site.

  Optional field values to set:

   string NewName: Name of the Site View. Max Length=50

   string Description: User description. Default="" Max Length=250
```

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsSiteView object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsSiteView for Individual Fields

Get the PvsSiteView into a $o variable. Change the $o field values and then Set the PvsSiteView with the result.

```
$o = Get-PvsSiteView -Name oldSiteView -SiteName theSite -Fields Name
$o.Name = "newSiteView"
Set-PvsSiteView $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Set PvsSiteView for a Field Using Pipe

Get the PvsSiteView into a $o variable. Change a $o field to the correct value and then Set the PvsSiteView with the result.

```
Get-PvsSiteView -Name oldSiteView -SiteName theSite -Fields Name |
foreach { $o = $_; $o.Name = "newSiteView"; $o } |
Set-PvsSiteView
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsStore

Set Store(s) changed values from PvsStore object(s), or set one or more field values for a PvsStore.

Required

PvsStore Store: PvsStore object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsStore object, and only will be set if the value has changed.

string Name or StoreName: Name of the Store. Max Length=50

Guid SiteId: GUID of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteName can be used instead. Default=00000000-0000-0000-0000-000000000000

string SiteName: Name of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteId can be used instead. Default=""

string Description: User description. Default="" Max Length=250

string Path: Default directory path that the Servers use to access this Store. Max Length=255

string[] CachePath: Default Cache path(s) that the Servers use with this Store. If none are specified the caches will be placed in the WriteCache subdirectory of the Store path. Default=None

When Store is not passed, the parameters below are used:

One of these required

Guid Guid or StoreId: GUID of the Store to Set.

string Name or StoreName: Name of the Store to Set.

Optional field values to set:

string NewName: Name of the Store. Max Length=50

Guid SiteId: GUID of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteName can be used instead. Default=00000000-0000-0000-0000-000000000000

string SiteName: Name of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteId can be used instead. Default=""

string Description: User description. Default="" Max Length=250

string Path: Default directory path that the Servers use to access this Store. Max Length=255

string[] CachePath: Default Cache path(s) that the Servers use with this Store. If none are specified the caches will be placed in the WriteCache subdirectory of the Store path. Default=None

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsStore object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsStore for Individual Fields

Get the PvsStore into a $o variable. Change the $o field values and then Set the PvsStore with the result.

```
$o = Get-PvsStore -Name theStore -Fields CachePath

$o.CachePath = "\\Network\CachePath"
```

```
Set-PvsStore $o

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 2: Set PvsStore for a Field Using Pipe

```
Get the PvsStore into a $o variable. Change a $o field to the correct
                value and then Set the PvsStore with the result.

Get-PvsStore -Name theStore -Fields CachePath | foreach { $o = $_;
                $o.CachePath = "\\Network\CachePath"; $o } | Set-
                PvsStore

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y
                and returns the object again so it can be piped to
                the Set command for update.
```

Set-PvsUpdateTask

```
Set Update Task(s) changed values from PvsUpdateTask object(s), or set one
  or more field values for a PvsUpdateTask.

   Required

    PvsUpdateTask UpdateTask: PvsUpdateTask object(s) with changed
                property value(s) to be set. The object(s) can come
                from a pileline.

    These values are in the PvsUpdateTask object, and only will be set if
                the value has changed.

    string Name or UpdateTaskName: Name of the Update Task. It is unique
                within the Site. Max Length=50

    string Description: User description. Default="" Max Length=250

    bool Enabled: True when it will be processed, false otherwise.
                Default=true

    uint Hour: The hour of the day to perform the task. Min=0, Max=23,
                Default=0

    uint Minute: The minute of the hour to perform the task. Min=0,
                Max=59, Default=0

    uint Recurrence: The update will reoccur on this schedule. 0 = None, 1
                = Daily, 2 = Every Weekday, 3 = Weekly, 4 = Monthly
                Date, 5 = Monthly Type. Min=0, Max=5, Default=0

    uint DayMask: Days selected values. 1 = Monday, 2 = Tuesday, 4 =
                Wednesday, 8 = Thursday, 16 = Friday, 32 = Saturday,
                64 = Sunday, 128 = Day. Default=0. This is used with
                Weekly and Monthly Type recurrence. Min=1, Max=255,
                Default=4

    uint[] Date: Days of the month. Numbers from 1-31 are the only valid
                values. This is used with Monthly Date recurrence.
                Default="" Max Length=83
```

uint MonthlyOffset: When to happen monthly. 0 = None, 1 = First, 2 = Second, 3 = Third, 4 = Forth, 5 = Last. This is used with Monthly Type recurrence. Min=0, Max=5, Default=3

string EsdType: Esd to use. Valid values are SCCM or WSUS. If no value, a custom script is run on the client. Default="" Max Length=50

string PreUpdateScript: Script file to run before the update starts. Default="" Max Length=255

string PreVmScript: Script file to run before the VM is loaded. Default="" Max Length=255

string PostUpdateScript: Script file to run after the update finishes. Default="" Max Length=255

string PostVmScript: Script file to run after the VM is unloaded. Default="" Max Length=255

string Domain: Domain to add the Disk Update Device(s) to. If not included, the first Domain Controller found on the Server is used. Default="" Max Length=255

string OrganizationUnit: Organizational Unit to add the Disk Update Device(s) to. This parameter is optional. If it is not specified, the device is added to the built in Computers container. Child OU's should be delimited with forward slashes, e.g. "ParentOU/ChildOU". Special characters in an OU name, such as '"', '#', '+', ',', ';', '>', '=', must be escaped with a backslash. For example, an OU called "commaIn,TheMiddle" must be specified as "commaIn\,TheMiddle". The old syntax of delimiting child OU's with a comma is still supported, but deprecated. Note that in this case, the child OU comes first, e.g. "ChildOU,ParentOU". Default="" Max Length=255

uint PostUpdateApprove: Access to place the version in after the update has occurred. 0 = Production, 1 = Test, 2 = Maintenance. Min=0, Max=2, Default=0

When UpdateTask is not passed, the parameters below are used:

This required

Guid Guid or UpdateTaskId: GUID of the Update Task to Set.

or this required & resolution

string Name or UpdateTaskName: Name of the Update Task to Set.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

Optional field values to set:

string NewName: Name of the Update Task. It is unique within the Site. Max Length=50

string Description: User description. Default="" Max Length=250

bool Enabled: True when it will be processed, false otherwise. Default=true

uint Hour: The hour of the day to perform the task. Min=0, Max=23, Default=0

uint Minute: The minute of the hour to perform the task. Min=0, Max=59, Default=0

uint Recurrence: The update will reoccur on this schedule. 0 = None, 1 = Daily, 2 = Every Weekday, 3 = Weekly, 4 = Monthly Date, 5 = Monthly Type. Min=0, Max=5, Default=0

uint DayMask: Days selected values. 1 = Monday, 2 = Tuesday, 4 = Wednesday, 8 = Thursday, 16 = Friday, 32 = Saturday, 64 = Sunday, 128 = Day. Default=0. This is used with Weekly and Monthly Type recurrence. Min=1, Max=255, Default=4

uint[] Date: Days of the month. Numbers from 1-31 are the only valid values. This is used with Monthly Date recurrence. Default="" Max Length=83

uint MonthlyOffset: When to happen monthly. 0 = None, 1 = First, 2 = Second, 3 = Third, 4 = Forth, 5 = Last. This is used with Monthly Type recurrence. Min=0, Max=5, Default=3

string EsdType: Esd to use. Valid values are SCCM or WSUS. If no value, a custom script is run on the client. Default="" Max Length=50

string PreUpdateScript: Script file to run before the update starts. Default="" Max Length=255

string PreVmScript: Script file to run before the VM is loaded. Default="" Max Length=255

string PostUpdateScript: Script file to run after the update finishes. Default="" Max Length=255

string PostVmScript: Script file to run after the VM is unloaded. Default="" Max Length=255

string Domain: Domain to add the Disk Update Device(s) to. If not included, the first Domain Controller found on the Server is used. Default="" Max Length=255

string OrganizationUnit: Organizational Unit to add the Disk Update Device(s) to. This parameter is optional. If it is not specified, the device is added to the built in Computers container. Child OU's should be delimited with forward slashes, e.g. "ParentOU/ChildOU". Special characters in an OU name, such as '"', '#', '+', ',', ';', '>', '=', must be escaped with a backslash. For example, an OU called "commaIn,TheMiddle" must be specified as "commaIn\,TheMiddle". The old syntax of delimiting child OU's with a comma is still supported, but deprecated. Note that in this case, the child OU comes first, e.g. "ChildOU,ParentOU". Default="" Max Length=255

uint PostUpdateApprove: Access to place the version in after the update has occurred. 0 = Production, 1 = Test, 2 = Maintenance. Min=0, Max=2, Default=0

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsUpdateTask object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsUpdateTask for Individual Fields

Get the PvsUpdateTask into a $o variable. Change the $o field values and then Set the PvsUpdateTask with the result.

```
$o = Get-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Fields Date, Recurrence

$o.Date = 1, 15

$o.Recurrence = 4

Set-PvsUpdateTask $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Set PvsUpdateTask for a Field Using Pipe

Get the PvsUpdateTask into a $o variable. Change a $o field to the correct value and then Set the PvsUpdateTask with the result.

```
Get-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Fields Recurrence | foreach { $o = $_; $o.Recurrence = 1; $o } | Set-PvsUpdateTask
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsVirtualHostingPool

Set Virtual Hosting Pool(s) changed values from PvsVirtualHostingPool object(s), or set one or more field values for a PvsVirtualHostingPool.

Required

PvsVirtualHostingPool VirtualHostingPool: PvsVirtualHostingPool object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsVirtualHostingPool object, and only will be set if the value has changed.

string Name or VirtualHostingPoolName: Name of the Virtual Hosting Pool. It is unique within the Site. Max Length=50

uint Type: Type of the Virtual Hosting Pool. 0 = Citrix XenServer, 1 = Microsoft SCVMM/Hyper-V, 2 = VMWare vSphere/ESX. Min=0, Max=3, Default=0

string Description: User description. Default="" Max Length=250

string Server: Name or IP of the Host Server. Max Length=255

uint Port: Port of the Host Server. Min=80, Max=65534, Default=80

string Datacenter: Datacenter of the Virtual Hosting Pool. Default="" Max Length=250

uint UpdateLimit: Number of updates at the same time. Min=2, Max=1000, Default=1000

uint UpdateTimeout: Timeout for updates. Min=2, Max=240, Default=60

uint ShutdownTimeout: Timeout for shutdown. Min=2, Max=30, Default=10

string UserName: Name to use when logging into the Server.

string Password: Password to use when logging into the Server.

Guid XdHostingUnitUuid: UUID of XenDesktop Hosting Unit Default=00000000-0000-0000-0000-000000000000

bool PrepopulateEnabled: Enable prepopulate when set to true Default=false

Guid XsPvsSiteUuid: UUID of XenServer PVS_site Default=00000000-0000-0000-0000-000000000000

string PlatformVersion: Hypervisor Host Version Default="" Max Length=250

string XdHcHypervisorConnectionName: Hypervisor Connection Name for HCL Connection Details object Default="" Max Length=250

string XdHcHypervisorConnectionUid: Hypervisor Connection Uid for HCL Connection Details object Default="" Max Length=250

string XdHcRevision: Revision for HCL Connection Details object Default="" Max Length=250

string XdHcCustomProperties: Custom Properties for HCL Connection Details object Default="" Max Length=250

string XdHcSslThumbprints: Ssl Thumbprints for HCL Connection Details object Default="" Max Length=250

string DisableHostXsProxy: True to disable PVS-Accelerator Default=""

When VirtualHostingPool is not passed, the parameters below are used:

This required

Guid Guid or VirtualHostingPoolId: GUID of the Virtual Hosting Pool to Set.

or this required & resolution

string Name or VirtualHostingPoolName: Name of the Virtual Hosting Pool to Set.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

Optional field values to set:

string NewName: Name of the Virtual Hosting Pool. It is unique within the Site. Max Length=50

uint Type: Type of the Virtual Hosting Pool. 0 = Citrix XenServer, 1 = Microsoft SCVMM/Hyper-V, 2 = VMWare vSphere/ESX. Min=0, Max=3, Default=0

string Description: User description. Default="" Max Length=250

string Server: Name or IP of the Host Server. Max Length=255

uint Port: Port of the Host Server. Min=80, Max=65534, Default=80

string Datacenter: Datacenter of the Virtual Hosting Pool. Default="" Max Length=250

uint UpdateLimit: Number of updates at the same time. Min=2, Max=1000, Default=1000

uint UpdateTimeout: Timeout for updates. Min=2, Max=240, Default=60

uint ShutdownTimeout: Timeout for shutdown. Min=2, Max=30, Default=10

string UserName: Name to use when logging into the Server.

string Password: Password to use when logging into the Server.

Guid XdHostingUnitUuid: UUID of XenDesktop Hosting Unit Default=00000000-0000-0000-0000-000000000000

bool PrepopulateEnabled: Enable prepopulate when set to true Default=false

Guid XsPvsSiteUuid: UUID of XenServer PVS_site Default=00000000-0000-0000-0000-000000000000

string PlatformVersion: Hypervisor Host Version Default="" Max Length=250

string XdHcHypervisorConnectionName: Hypervisor Connection Name for HCL Connection Details object Default="" Max Length=250

string XdHcHypervisorConnectionUid: Hypervisor Connection Uid for HCL Connection Details object Default="" Max Length=250

string XdHcRevision: Revision for HCL Connection Details object Default="" Max Length=250

string XdHcCustomProperties: Custom Properties for HCL Connection Details object Default="" Max Length=250

string XdHcSslThumbprints: Ssl Thumbprints for HCL Connection Details object Default="" Max Length=250

string DisableHostXsProxy: True to disable PVS-Accelerator Default=""

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsVirtualHostingPool object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsVirtualHostingPool for Individual Fields

Get the PvsVirtualHostingPool into a $o variable. Change the $o field values and then Set the PvsVirtualHostingPool with the result.

```
$o = Get-PvsVirtualHostingPool -Name theVirtualHostingPool -SiteName theSite -Fields Port

$o.Port = 180

Set-PvsVirtualHostingPool $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Set PvsVirtualHostingPool for a Field Using Pipe

Get the PvsVirtualHostingPool into a $o variable. Change a $o field to the correct value and then Set the PvsVirtualHostingPool with the result.

```
Get-PvsVirtualHostingPool -Name theVirtualHostingPool -SiteName theSite -Fields Port | foreach { $o = $_; $o.Port = 180; $o } | Set-PvsVirtualHostingPool
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Set-PvsXDSite

Set XDSite(s) changed values from PvsXDSite object(s), or set one or more field values for a PvsXDSite.

Required

PvsXDSite XDSite: PvsXDSite object(s) with changed property value(s) to be set. The object(s) can come from a pileline.

These values are in the PvsXDSite object, and only will be set if the value has changed.

string[] ConfigServices: XenDesktop Server addresses. Max Length=2000

When XDSite is not passed, the parameters below are used:

This required

Guid Guid or XdSiteId: GUID of the XenDesktop Site to Set.

Optional field values to set:

string[] ConfigServices: XenDesktop Server addresses. Max Length=2000

Optional

SwitchParameter PassThru: If -PassThru is specified, the resulting PvsXDSite object(s) are returned.

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Set PvsXDSite for Individual Fields

Get the PvsXDSite into a $o variable. Change the $o field values and then Set the PvsXDSite with the result.

```
$o = Get-PvsXDSite -Guid "27965b92-7034-408c-9910-cfff53f1feec" -
Fields ConfigServices

$o.ConfigServices = "192.168.0.190", "192.168.0.191"

Set-PvsXDSite $o
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 2: Set PvsXDSite for a Field Using Pipe

Get the PvsXDSite into a $o variable. Change a $o field to the correct value and then Set the PvsXDSite with the result.

```
Get-PvsXDSite -Guid "27965b92-7034-408c-9910-cfff53f1feec" -Fields
ConfigServices | foreach { $o = $_; $o.ConfigServices
= "192.168.0.190", "192.168.0.191"; $o } | Set-
PvsXDSite
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

The "foreach { $o = $_; $o.X = Y; $o }" sets the field X to value Y and returns the object again so it can be piped to the Set command for update.

Start-PvsAutoUpdate

Apply Auto Update for a Server or all Servers in a Site.

One of these required

Guid[] Guid or ServerId: GUID of the Server to apply Auto Update.

string[] Name or ServerName: Name of the Server to apply Auto Update.

Guid[] SiteId: GUID of the Site to apply Auto Update on all Servers.

string[] SiteName: Name of the Site to apply Auto Update on all Servers.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId or SiteId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Start-PvsAutoUpdate for Name

```
Start-PvsAutoUpdate -Name theServer
```

EXAMPLE 2: Start-PvsAutoUpdate for PvsServer Using Pipe

The Get-PvsServer output is piped to the Start-PvsAutoUpdate.

```
Get-PvsServer -Name theServer -Fields -ServerId | Start-PvsAutoUpdate
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Start-PvsAutoUpdate for SiteName

```
Start-PvsAutoUpdate -SiteName theSite
```

EXAMPLE 4: Start-PvsAutoUpdate for PvsSite Using Pipe

The Get-PvsSite output is piped to the Start-PvsAutoUpdate.

```
Get-PvsSite -SiteName theSite -Fields -SiteId | Start-PvsAutoUpdate
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Start-PvsCreateDisk

Create a Disk and the Disk Locator for it. Returns the PvsDiskLocator created if finished. If not returned, then call CreateDiskStatus to get the PvsDiskLocator when processing finishes.

This required

string Name: Name of the Disk file to be created.

This required

UInt64 Size: Size of the disk in Megabytes for a fixed size disk. Maximum size in Megabytes for a dynamically sized disk. Min=1, VHD Max=2088960, VHDX Max=67108864

One of these required

```
    Guid StoreId: GUID of the Store that the Disk will be a member of.

    string StoreName: Name of the Store that the Disk will be a member of.

  One of these required

    Guid SiteId: GUID of the Site.

    string SiteName: Name of the Site.

  One of these optional

    Guid ServerId: GUID of the only Server for this Disk.

    string ServerName: Name of the only Server for this Disk.

  Optional

    string Description: Description of the Disk that will be placed in the
                        Disk Locator.

    SwitchParameter CreateDiskDisabled: If -CreateDiskDisabled is
                        specified, the Disk will be created disabled. It is
                        created Enabled by default.

    SwitchParameter VHDX: If -VHDX is specified, VHDX will be used for the
                        format of the image. VHDX has a Block size of 32 MB.
                        VHD is the default.

    SwitchParameter Dynamic: If -Dynamic is specified, a dynamic VHD that
                        will be created. The default is fixed.

    uint VhdBlockSize: Block size in KB. For VHD it is only used with
                        Dynamic type. Tested sizes for VHD are 512, 2048, and
                        16384. VHD Min=512, Max=16384, Default=2048. For VHDX
                        it is used for all types. Tested size for VHDX is
                        32768. VHDX Min=1024, Max= 262144, Default=32768.

    uint LogicalSectorSize: Only used with VHDX format. Logical Sector
                        Size. Values are: 512, 4096, Default=512

  Instead of a parameter that matches one of the members listed

    PvsObject[] Object: PvsObjects with the members below can be used as
                        the Object parameter or from a pileline:

      StoreId, SiteId or ServerId

  Optional

    SwitchParameter Confirm: The impact of this operation is "low". If -
                        Confirm is specified, the operation will be
                        confirmed. $ConfirmPreference can be set to "low" to
                        have confirmation without the Confirm parameter.

  PvsDiskLocator: If the create is already finished and successful, the
    PvsDiskLocator is returned.
```

EXAMPLE 1: Start-PvsCreateDisk for VHDX to PvsDiskLocator

```
    This example creates a Dynamic VHDX Disk that has a size of 20 GB.

    $thePvsDiskLocator = Start-PvsCreateDisk -Name theDiskName -Size 20480
                        -StoreName theStore -SiteName theSite -VHDX -Dynamic

    while ($thePvsDiskLocator -eq $null)                              # while
                        the create is processing
```

```
{
     %percentFinished = Get-PvsCreateDiskStatus -Name theDiskName -
                        StoreName theStore  # get percent finished or
                        DiskLocator when done
     if (%percentFinished.GetType().Name == "PvsDiskLocator")
     {
          $thePvsDiskLocator = %percentFinished
     }
     else
     {
          %percentFinished.ToString() + "% finished"                #
                             display percent finished
          Start-Sleep -seconds 10                                   #
                             wait 10 seconds more
     }
}
"Successful"
```

EXAMPLE 2: Start-PvsCreateDisk for VHD to PvsDiskLocator

```
TThis example creates a Fixed VHD Disk that has a size of 20 GB.
$thePvsDiskLocator = Start-PvsCreateDisk -Name theDiskName -Size 20480
                  -StoreName theStore -SiteName theSite
while ($thePvsDiskLocator -eq $null)                              # while
                  the create is processing
{
     %percentFinished = Get-PvsCreateDiskStatus -Name theDiskName -
                        StoreName theStore  # get percent finished or
                        DiskLocator when done
     if (%percentFinished.GetType().Name == "PvsDiskLocator")
     {
          $thePvsDiskLocator = %percentFinished
     }
     else
     {
          %percentFinished.ToString() + "% finished"                #
                             display percent finished
          Start-Sleep -seconds 10                                   #
                             wait 10 seconds more
     }
}
"Successful"
```

Start-PvsDeviceBoot

Boot a Device, Collection or View. Returns a PvsTask of the Task being run. With the PvsTask, call Get-PvsTaskStatus to get the percent complete, Get-PvsTask to get the results, and Stop-PvsTask to stop it early. The PvsTask Results has the DeviceName/value of the devices that succeeded in the first name/value pair of each record of the XML.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Boot.

string[] Name or DeviceName: Name of the Device to Boot.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Boot.

Guid CollectionId: GUID of the Collection to Boot all Devices.

Guid SiteViewId: GUID of the Site View to Boot all Devices.

Guid FarmViewId: GUID of the Farm View to Boot all Devices.

string FarmViewName: Name of the Farm View to Boot all Devices.

or one of these required & resolutions

string CollectionName: Name of the Collection to Boot all Devices.

string SiteViewName: Name of the Site View to Boot all Devices.

One of these resolutions when needed

Guid SiteId: GUID of the Site.

string SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceName, CollectionId, SiteViewId or FarmViewId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsTask: If successful, the PvsTask object for the process started is returned.

EXAMPLE 1: Start-PvsDeviceBoot for DeviceName to PvsTask

```
$thePvsTask = Start-PvsDeviceBoot -DeviceName theDevice             #
                start the task
while ($thePvsTask.State -eq 0)                                   # while
                the task is processing
{
     %percentFinished = Get-PvsTaskStatus -Object $thePvsTask   # get
                percent finished
     %percentFinished.ToString() + "% finished"                  #
                display percent finished
```

```
        Start-Sleep -seconds 10                                          # wait
                        10 seconds more

        $thePvsTask = Get-PvsTask -Object $thePvsTask                    # get
                        the current PvsTask object

    }

    if ($thePvsTask.State -eq 2)                                        # check
                    to see if completed

    {

        "Successful"

    }

    else

    {

        "Cancelled"

    }
```

EXAMPLE 2: Start-PvsDeviceBoot for DeviceMac

```
    Start-PvsDeviceBoot -DeviceMac "00-11-22-33-44-55"
```

EXAMPLE 3: Start-PvsDeviceBoot for FarmViewName

```
    Start-PvsDeviceBoot -FarmViewName theFarmView
```

EXAMPLE 4: Start-PvsDeviceBoot for CollectionName

```
    Start-PvsDeviceBoot -CollectionName theCollection -SiteName theSite

    CollectionId can be used instead of CollectionName so that the
                    SiteName or SiteId is not also needed.
```

EXAMPLE 5: Start-PvsDeviceBoot for SiteViewName

```
    Start-PvsDeviceBoot -SiteViewName theSiteView -SiteName theSite

    SiteViewId can be used instead of SiteViewName so that the SiteName or
                    SiteId is not also needed.
```

Start-PvsDeviceDiskTempVersionMode

```
Set a Temporary Disk Version for the specified Device and optional
  DiskLocator. The Temporary Disk Version uses the production version that
  all production Devices boot from. Once set, the Device boots this
  Temporary Disk Version instead of any currently assigned vDisk. Not
  supported for non-production and Personal vDisk Devices. Cannot be done
  when the the Device already has a Temporary Disk Version, when the
  DiskLocator is using server side persistent cache mode or the active
  production version is in private mode.

   One of these required

    Guid[] Guid or DeviceId: GUID of the Device to start to use a
                    temporary disk version.

    string[] Name or DeviceName: Name of the Device to start to use a
                    temporary disk version.
```

PvsPhysicalAddress[] DeviceMac: MAC of the Device to start to use a temporary disk version.

This optional

Guid[] DiskLocatorId: GUID of the Disk Locator to use a temporary version of. If not specified, the only Disk Locator assigned to the Device is used.

or this optional & resolution

string[] DiskLocatorName: Name of the Disk Locator to use a temporary version of. If not specified, the only Disk Locator assigned to the Device is used.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId or DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Start-PvsDeviceDiskTempVersionMode for DeviceName with DiskLocatorName

```
Start-PvsDeviceDiskTempVersionMode -Name theDevice -DiskLocatorName
theDiskLocator -StoreName theStore
```

Start-PvsDeviceReboot

Reboot a Device, Collection or View. Returns a PvsTask of the Task being run. With the PvsTask, call Get-PvsTaskStatus to get the percent complete, Get-PvsTask to get the results, and Stop-PvsTask to stop it early. The PvsTask Results has the DeviceName/value of the devices that succeeded in the first name/value pair of each record of the XML.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to Reboot.

string[] Name or DeviceName: Name of the Device to Reboot.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to Reboot.

Guid CollectionId: GUID of the Collection to Reboot all Devices.

Guid SiteViewId: GUID of the Site View to Reboot all Devices.

Guid FarmViewId: GUID of the Farm View to Reboot all Devices.

string FarmViewName: Name of the Farm View to Reboot all Devices.

Guid DiskLocatorId: GUID of the DiskLocator to Reboot all Devices.

```
or one of these required & resolutions
 string CollectionName: Name of the Collection to Reboot all Devices.
 string SiteViewName: Name of the Site View to Reboot all Devices.
 string DiskLocatorName: Name of the DiskLocator to Reboot all Devices.
Optional
 string Message: Message to display before rebooting the Device(s).
                 Default=The target device is being shutdown remotely
                 by the Console.
 uint Delay: Number of seconds to delay before rebooting the Device(s).
                 Default=10
 uint Version: Version of the Disk to Reboot all Devices for. This is
                 used with DiskLocatorId or DiskLocatorName.
One of these resolutions when needed
 Guid SiteId: GUID of the Site.
 string SiteName: Name of the Site.
One of these resolutions when needed
 Guid StoreId: GUID of the Store that is needed when a DiskLocatorName
                 is used.
 string StoreName: Name of the Store that is needed when a
                 DiskLocatorName is used.
Instead of a parameter that matches one of the members listed
 PvsObject[] Object: PvsObjects with the members below can be used as
                 the Object parameter or from a pileline:
  DeviceName, CollectionId, SiteViewId, FarmViewId or DiskLocatorId
Optional
 SwitchParameter Confirm: The impact of this operation is "medium". If
                 -Confirm is specified, the operation will be
                 confirmed. $ConfirmPreference can be set to "medium"
                 or "low" to have confirmation without the Confirm
                 parameter.
PvsTask: If successful, the PvsTask object for the process started is
  returned.
```

EXAMPLE 1: Start-PvsDeviceReboot for DeviceName to PvsTask

```
 $thePvsTask = Start-PvsDeviceReboot -DeviceName theDevice            #
                 start the task
 while ($thePvsTask.State -eq 0)                                    # while
                 the task is processing
 {
      %percentFinished = Get-PvsTaskStatus -Object $thePvsTask   # get
                      percent finished
      %percentFinished.ToString() + "% finished"                  #
                      display percent finished
```

```
        Start-Sleep -seconds 10                                          # wait
                        10 seconds more

        $thePvsTask = Get-PvsTask -Object $thePvsTask                    # get
                        the current PvsTask object

    }

    if ($thePvsTask.State -eq 2)                                        # check
                    to see if completed

    {

        "Successful"

    }

    else

    {

        "Cancelled"

    }
```

EXAMPLE 2: Start-PvsDeviceReboot for DeviceMac

```
    Start-PvsDeviceReboot -DeviceMac "00-11-22-33-44-55"
```

EXAMPLE 3: Start-PvsDeviceReboot for FarmViewName

```
    Start-PvsDeviceReboot -FarmViewName theFarmView
```

EXAMPLE 4: Start-PvsDeviceReboot for CollectionName

```
    Start-PvsDeviceReboot -CollectionName theCollection -SiteName theSite

    CollectionId can be used instead of CollectionName so that the
                    SiteName or SiteId is not also needed.
```

EXAMPLE 5: Start-PvsDeviceReboot for SiteViewName

```
    Start-PvsDeviceReboot -SiteViewName theSiteView -SiteName theSite

    SiteViewId can be used instead of SiteViewName so that the SiteName or
                    SiteId is not also needed.
```

Start-PvsDeviceShutdown

```
Shutdown a Device, Collection or View. Returns a PvsTask of the Task being
    run. With the PvsTask, call Get-PvsTaskStatus to get the percent
    complete, Get-PvsTask to get the results, and Stop-PvsTask to stop it
    early. The PvsTask Results has the DeviceName/value of the devices that
    succeeded in the first name/value pair of each record of the XML.

     One of these required

      Guid[] Guid or DeviceId: GUID of the Device to Shutdown.

      string[] Name or DeviceName: Name of the Device to Shutdown.

      PvsPhysicalAddress[] DeviceMac: MAC of the Device to Shutdown.

      Guid CollectionId: GUID of the Collection to Shutdown all Devices.

      Guid SiteViewId: GUID of the Site View to Shutdown all Devices.

      Guid FarmViewId: GUID of the Farm View to Shutdown all Devices.
```

```
 string FarmViewName: Name of the Farm View to Shutdown all Devices.

 Guid DiskLocatorId: GUID of the DiskLocator to Shutdown all Devices.

or one of these required & resolutions

 string CollectionName: Name of the Collection to Shutdown all Devices.

 string SiteViewName: Name of the Site View to Shutdown all Devices.

 string DiskLocatorName: Name of the DiskLocator to Shutdown all
                 Devices.

Optional

 string Message: Message to display before shutting down the Device(s).
                 Default=The target device is being shutdown remotely
                 by the Console.

 uint Delay: Number of seconds to delay before shutting down the
                 Device(s). Default=10

 uint Version: Version of the Disk to Shutdown all Devices for. This is
                 used with DiskLocatorId or DiskLocatorName.

One of these resolutions when needed

 Guid SiteId: GUID of the Site.

 string SiteName: Name of the Site.

One of these resolutions when needed

 Guid StoreId: GUID of the Store that is needed when a DiskLocatorName
                 is used.

 string StoreName: Name of the Store that is needed when a
                 DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

 PvsObject[] Object: PvsObjects with the members below can be used as
                 the Object parameter or from a pileline:

   DeviceName, CollectionId, SiteViewId, FarmViewId or DiskLocatorId

Optional

 SwitchParameter Confirm: The impact of this operation is "medium". If
                 -Confirm is specified, the operation will be
                 confirmed. $ConfirmPreference can be set to "medium"
                 or "low" to have confirmation without the Confirm
                 parameter.

PvsTask: If successful, the PvsTask object for the process started is
   returned.
```

EXAMPLE 1: Start-PvsDeviceShutdown for DeviceName to PvsTask

```
 $thePvsTask = Start-PvsDeviceShutdown -DeviceName theDevice          #
                 start the task

 while ($thePvsTask.State -eq 0)                                    # while
                 the task is processing

 {

      %percentFinished = Get-PvsTaskStatus -Object $thePvsTask   # get
                      percent finished
```

```
        %percentFinished.ToString() + "% finished"                  #
                        display percent finished

        Start-Sleep -seconds 10                                     # wait
                        10 seconds more

        $thePvsTask = Get-PvsTask -Object $thePvsTask               # get
                        the current PvsTask object

    }

    if ($thePvsTask.State -eq 2)                                    # check
                    to see if completed

    {

        "Successful"

    }

    else

    {

        "Cancelled"

    }
```

EXAMPLE 2: Start-PvsDeviceShutdown for DeviceMac

```
    Start-PvsDeviceShutdown -DeviceMac "00-11-22-33-44-55"
```

EXAMPLE 3: Start-PvsDeviceShutdown for FarmViewName

```
    Start-PvsDeviceShutdown -FarmViewName theFarmView
```

EXAMPLE 4: Start-PvsDeviceShutdown for CollectionName

```
    Start-PvsDeviceShutdown -CollectionName theCollection -SiteName
                    theSite

    CollectionId can be used instead of CollectionName so that the
                    SiteName or SiteId is not also needed.
```

EXAMPLE 5: Start-PvsDeviceShutdown for SiteViewName

```
    Start-PvsDeviceShutdown -SiteViewName theSiteView -SiteName theSite

    SiteViewId can be used instead of SiteViewName so that the SiteName or
                    SiteId is not also needed.
```

Start-PvsDeviceUpdateBdm

```
Update the BDM partition for a Device, Collection or View. Returns a
    PvsTask of the Task being run. With the PvsTask, call Get-PvsTaskStatus
    to get the percent complete, Get-PvsTask to get the results, and Stop-
    PvsTask to stop it early. The PvsTask Results has the DeviceName/value of
    the devices that succeeded in the first name/value pair of each record of
    the XML.

     One of these required

      Guid[] Guid or DeviceId: GUID of the Device to Update.

      string[] Name or DeviceName: Name of the Device to Update.

      Guid CollectionId: GUID of the Collection to Update all BDM Devices.
```

```
 Guid SiteViewId: GUID of the Site View to Update all BDM Devices.

 Guid FarmViewId: GUID of the Farm View to Update all BDM Devices.

 string FarmViewName: Name of the Farm View to Update all BDM Devices.

or one of these required & resolutions

 string CollectionName: Name of the Collection to Update all BDM
                 Devices.

 string SiteViewName: Name of the Site View to Update all BDMDevices.

One of these resolutions when needed

 Guid SiteId: GUID of the Site.

 string SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

 PvsObject[] Object: PvsObjects with the members below can be used as
                 the Object parameter or from a pileline:

  DeviceName, CollectionId, SiteViewId or FarmViewId

Optional

 SwitchParameter Confirm: The impact of this operation is "low". If -
                 Confirm is specified, the operation will be
                 confirmed. $ConfirmPreference can be set to "low" to
                 have confirmation without the Confirm parameter.

PvsTask: If successful, the PvsTask object for the process started is
  returned.
```

EXAMPLE 1: Start-PvsDeviceUpdateBdm for DeviceName to PvsTask

```
 $thePvsTask = Start-PvsDeviceUpdateBdm -DeviceName
                 theDevice              # start the task

 while ($thePvsTask.State -eq 0)                                    # while
                 the task is processing

 {

      %percentFinished = Get-PvsTaskStatus -Object $thePvsTask   # get
                      percent finished

      %percentFinished.ToString() + "% finished"                 #
                      display percent finished

      Start-Sleep -seconds 10                                    # wait
                      10 seconds more

      $thePvsTask = Get-PvsTask -Object $thePvsTask              # get
                      the current PvsTask object

 }

 if ($thePvsTask.State -eq 2)                                       # check
                 to see if completed

 {

      "Successful"

 }

 else
```

```
{
     "Cancelled"
}
```

EXAMPLE 2: Start-PvsDeviceUpdateBdm for FarmViewName

```
Start-PvsDeviceUpdateBdm -FarmViewName theFarmView
```

EXAMPLE 3: Start-PvsDeviceUpdateBdm for CollectionName

```
Start-PvsDeviceUpdateBdm -CollectionName theCollection -SiteName
                theSite

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 4: Start-PvsDeviceUpdateBdm for SiteViewName

```
Start-PvsDeviceUpdateBdm -SiteViewName theSiteView -SiteName theSite

SiteViewId can be used instead of SiteViewName so that the SiteName or
                SiteId is not also needed.
```

Start-PvsDisplayMessage

```
Display a message on a Device, Collection or View. Returns a PvsTask of the
  Task being run. With the PvsTask, call Get-PvsTaskStatus to get the
  percent complete, Get-PvsTask to get the results, and Stop-PvsTask to
  stop it early. The PvsTask Results has the DeviceName/value of the
  devices that succeeded in the first name/value pair of each record of the
  XML.

   This required

    string Message: Message to display on the Device(s).

   One of these required

    Guid[] DeviceId: GUID of the Device to Display a Message.

    string[] DeviceName: Name of the Device to Display a Message.

    PvsPhysicalAddress[] DeviceMac: MAC of the Device to Display a
                    Message.

    Guid CollectionId: GUID of the Collection to Display a Message on all
                    Devices.

    Guid SiteViewId: GUID of the Site View to Display a Message all
                    Devices.

    Guid FarmViewId: GUID of the Farm View to Display a Message on all
                    Devices.

    string FarmViewName: Name of the Farm View to Display a Message on all
                    Devices.

   or one of these required & resolutions

    string CollectionName: Name of the Collection to Display a Message on
                    all Devices.

    string SiteViewName: Name of the Site View to Display a Message on all
                    Devices.
```

```
One of these resolutions when needed

 Guid SiteId: GUID of the Site.

 string SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

 PvsObject[] Object: PvsObjects with the members below can be used as
                 the Object parameter or from a pileline:

  DeviceName, CollectionId, SiteViewId or FarmViewId

Optional

 SwitchParameter Confirm: The impact of this operation is "low". If -
                 Confirm is specified, the operation will be
                 confirmed. $ConfirmPreference can be set to "low" to
                 have confirmation without the Confirm parameter.

PvsTask: If successful, the PvsTask object for the process started is
  returned.
```

EXAMPLE 1: Start-PvsDisplayMessage for DeviceName to PvsTask

```
$thePvsTask = Start-PvsDisplayMessage -DeviceName theDevice -Message
                "message" # start the task

while ($thePvsTask.State -eq 0)                                    # while
                the task is processing

{

     %percentFinished = Get-PvsTaskStatus -Object $thePvsTask   # get
                     percent finished

     %percentFinished.ToString() + "% finished"                 #
                     display percent finished

     Start-Sleep -seconds 10                                    # wait
                     10 seconds more

     $thePvsTask = Get-PvsTask -Object $thePvsTask              # get
                     the current PvsTask object

}

if ($thePvsTask.State -eq 2)                                       # check
                to see if completed

{

     "Successful"

}

else

{

     "Cancelled"

}
```

EXAMPLE 2: Start-PvsDisplayMessage for DeviceMac

```
Start-PvsDisplayMessage -DeviceMac "00-11-22-33-44-55" -Message
                "message"
```

EXAMPLE 3: Start-PvsDisplayMessage for FarmViewName

```
Start-PvsDisplayMessage -FarmViewName theFarmView -Message "message"
```

EXAMPLE 4: Start-PvsDisplayMessage for CollectionName

```
Start-PvsDisplayMessage -CollectionName theCollection -SiteName
                theSite -Message "message"

CollectionId can be used instead of CollectionName so that the
                SiteName or SiteId is not also needed.
```

EXAMPLE 5: Start-PvsDisplayMessage for SiteViewName

```
Start-PvsDisplayMessage -SiteViewName theSiteView -SiteName theSite -
                Message "message"

SiteViewId can be used instead of SiteViewName so that the SiteName or
                SiteId is not also needed.
```

Start-PvsReportBug

```
Report a bug at individual server level or at site level. Problem report
  can be uploaded to Citrix Systems or can be saved locally in a path
  accessible from all servers in the farm.

   One of these required

    Guid Guid or SiteId: GUID of the Site.

    Guid ServerId: GUID of the Server.

   or one of these required & resolutions

    string Name or SiteName: Name of the Site.

    string ServerName: Name of the Server.

   This required

    string Summary: Short summary describing the problem.

   This optional

    string SrNumber: Service Request number of the reported problem.

   This optional

    string Description: Description of the reported problem.

   This optional

    string Path: Path where problem report bundle is saved.

   This optional

    DateTime DateTime: DateTime around which the reported problem occured.

   Instead of a parameter that matches one of the members listed

    PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

     SiteId or ServerId

   Optional

    SwitchParameter Confirm: The impact of this operation is "low". If -
                Confirm is specified, the operation will be
```

confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

PvsTask: If successful, the PvsTask object for the process started is returned.

EXAMPLE 1: Start-PvsReportBug for SiteName to PvsTask

```
$thePvsTask = Start-PvsReportBug -SiteName theSite -bugReportSummary
                test bug report            # start the task

while ($thePvsTask.State -eq
                0)
                                  # while the task is processing

{

     %percentFinished = Get-PvsTaskStatus -Object
                     $thePvsTask
                     # get percent finished

     %percentFinished.ToString() + "%
                     finished"
                               # display percent finished

     Start-Sleep -seconds
                     10
                                                # wait 10 seconds more

     $thePvsTask = Get-PvsTask -Object
                     $thePvsTask
                               # get the current PvsTask object

}

if ($thePvsTask.State -eq
                2)
                                     # check to see if completed

{

     "Successful"

}

else

{

     "Cancelled"

}
```

Start-PvsStreamService

Start the Stream Service on a Server or all Servers in a Site.

One of these required

Guid[] Guid or ServerId: GUID of the Server to start the Stream Service.

string[] Name or ServerName: Name of the Server to start the Stream Service.

Guid[] SiteId: GUID of the Site to start the Stream Service on all Servers.

string[] SiteName: Name of the Site to start the Stream Service on all Servers.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId or SiteId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Start-PvsStreamService for Name

```
Start-PvsStreamService -Name theServer
```

EXAMPLE 2: Start-PvsStreamService for PvsServer Using Pipe

The Get-PvsServer output is piped to the Start-PvsStreamService.

```
Get-PvsServer -Name theServer -Fields Guid | Start-PvsStreamService
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Start-PvsStreamService for SiteName

```
Start-PvsStreamService -SiteName theSite
```

EXAMPLE 4: Start-PvsStreamService for PvsSite Using Pipe

The Get-PvsSite output is piped to the Start-PvsStreamService.

```
Get-PvsSite -Name theSite -Fields Guid | Start-PvsStreamService
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Start-PvsUpdateTask

Starts an Update Task.

This required

Guid[] Guid or UpdateTaskId: GUID of the Update Task to Start.

or this required & resolution

string[] Name or UpdateTaskName: Name of the Update Task to Start.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

UpdateTaskId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Start-PvsUpdateTask for Name

```
Start-PvsUpdateTask -Name theUpdateTask -SiteName theSite
```

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

EXAMPLE 2: Start-PvsUpdateTask for PvsUpdateTask Using Pipe

The Get-PvsUpdateTask output is piped to the Start-PvsUpdateTask.

```
Get-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Fields Guid |
Start-PvsUpdateTask
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

Stop-PvsCreateDisk

Cancel an active CreateDisk.

This required

string[] Name: Name of the Disk file that is being created.

One of these required

Guid[] StoreId: GUID of the Store that the Disk will be a member of.

string[] StoreName: Name of the Store that the Disk will be a member of.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

StoreId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Stop-PvsCreateDisk for Name

```
Stop-PvsCreateDisk -Name theDiskFile -StoreName theStore
```

EXAMPLE 2: Stop-PvsCreateDisk for PvsStore Using Pipe

The Get-PvsStore output is piped to the Stop-PvsCreateDisk.

Get-PvsStore -Name theStore -Fields Guid | Stop-PvsCreateDisk -Name theDiskFile

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Stop-PvsDeviceDiskTempVersionMode

Unset the Temporary Disk Version for the specified Device. The Device must not currently be booted. Once removed, the Device again uses any currently assigned vDisk when booted.

One of these required

Guid[] Guid or DeviceId: GUID of the Device to no longer use a temporary disk version for.

string[] Name or DeviceName: Name of the Device to no longer use a temporary disk version for.

PvsPhysicalAddress[] DeviceMac: MAC of the Device to no longer use a temporary disk version for.

Guid[] DiskLocatorId: GUID of the DiskLocator to no longer use temporary disk versions for.

Guid[] SiteId: GUID of the Site for resolution of the DiskLocatorName.

string[] SiteName: Name of the Site for resolution of the DiskLocatorName.

or this required & resolution

string[] DiskLocatorName: Name of the DiskLocator to no longer use temporary disk versions for.

This optional & resolution

string[] Version: Version of the DiskLocator to no longer use temporary disk versions for.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site for resolution of the DiskLocatorName.

string[] SiteName: Name of the Site for resolution of the DiskLocatorName.

One of these resolutions when needed

Guid[] DiskLocatorId: GUID of the DiskLocator to no longer use temporary disk versions for.

string[] DiskLocatorName: Name of the DiskLocator to no longer use temporary disk versions for.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DeviceId, DiskLocatorId or SiteId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Stop-PvsDeviceDiskTempVersionMode for Device

```
Stop-PvsDeviceDiskTempVersionMode -Name theDevice
```

EXAMPLE 2: Stop-PvsDeviceDiskTempVersionMode for DiskLocator

```
Stop-PvsDeviceDiskTempVersionMode -DiskLocatorName theDiskLocator -
SiteName theSite -StoreName theStore
```

DiskLocatorId can be used instead of DiskLocatorName so that the StoreName or StoreId are not also needed.

EXAMPLE 3: Stop-PvsDeviceDiskTempVersionMode with DiskLocator

```
Stop-PvsDeviceDiskTempVersionMode -DiskLocatorName theDiskLocator -
Version 4 -SiteName theSite -StoreName theStore
```

DiskLocatorId can be used instead of DiskLocatorName so that the StoreName or StoreId are not also needed.

Stop-PvsStreamService

Stop the Stream Service on a Server or all Servers in a Site.

One of these required

Guid[] Guid or ServerId: GUID of the Server to stop the Stream Service.

string[] Name or ServerName: Name of the Server to stop the Stream Service.

Guid[] SiteId: GUID of the Site to stop the Stream Service on all Servers.

string[] SiteName: Name of the Site to stop the Stream Service on all Servers.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId or SiteId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Stop-PvsStreamService for Name

```
Stop-PvsStreamService -Name theServer
```

EXAMPLE 2: Stop-PvsStreamService for PvsServer Using Pipe

```
The Get-PvsServer output is piped to the Stop-PvsStreamService.

Get-PvsServer -Name theServer -Fields Guid | Stop-PvsStreamService

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

EXAMPLE 3: Stop-PvsStreamService for SiteName

```
Stop-PvsStreamService -SiteName theSite
```

EXAMPLE 4: Stop-PvsStreamService for PvsSite Using Pipe

```
The Get-PvsSite output is piped to the Stop-PvsStreamService.

Get-PvsSite -Name theSite -Fields Guid | Stop-PvsStreamService

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.
```

Stop-PvsTask

```
Cancel a running Task.

    This required

     uint TaskId: Id of the Task to Cancel.

    Instead of a parameter that matches one of the members listed

     PvsObject[] Object: PvsObjects with the members below can be used as
                the Object parameter or from a pileline:

      TaskId

    Optional

     SwitchParameter Confirm: The impact of this operation is "medium". If
                -Confirm is specified, the operation will be
                confirmed. $ConfirmPreference can be set to "medium"
                or "low" to have confirmation without the Confirm
                parameter.
```

EXAMPLE 1: Stop-PvsTask for TaskId

```
Stop-PvsTask -TaskId 101
```

Stop-PvsUpdateTask

```
Cancel an active Update Task for an Update Device.

    This required

     Guid[] Guid or UpdateTaskId: GUID of the Update Task to Cancel.

    or this required & resolution

     string[] Name or UpdateTaskName: Name of the Update Task to Cancel.
```

One of these required

Guid[] DeviceId: GUID of a specific Update Device to Cancel an Update Task for.

string[] DeviceName: Name of a specific Update Device to Cancel an Update Task for.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

UpdateTaskId or DeviceId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Stop-PvsUpdateTask for Name

```
Stop-PvsUpdateTask -Name theUpdateTask -DeviceName theDevice
```

EXAMPLE 2: Stop-PvsUpdateTask for PvsUpdateTask Using Pipe

The Get-PvsUpdateTask output is piped to the Stop-PvsUpdateTask.

```
Get-PvsUpdateTask -Name theUpdateTask -SiteName theSite -Fields Guid |
Stop-PvsUpdateTask -DeviceName theDevice
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId is not also needed.

EXAMPLE 3: Stop-PvsUpdateTask for PvsDiskUpdateDevice Using Pipe

The Get-PvsDiskUpdateDevice output is piped to the Stop-PvsUpdateTask.

```
Get-PvsDiskUpdateDevice -Name theDevice -Fields Guid | Stop-
PvsUpdateTask -Name theUpdateTask
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Test-PvsDirectory

Validate a Directory on the Server specified.

One of these required

Guid[] Guid or ServerId: GUID of the Server to validate a Directory on.

string[] Name or ServerName: Name of the Server to validate a Directory on.

This required

string[] Path: Path of the Directory to validate.

This optional

SwitchParameter ReadOnly: If -ReadOnly is specified, the directory should be validated for a managed read-only store.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

ServerId

Optional

SwitchParameter Confirm: The impact of this operation is "low". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Test-PvsDirectory for Name

```
Test-PvsDirectory -Name theServer -Path "C:\directory\subdirectory"
```

EXAMPLE 2: Test-PvsDirectory for PvsServer Using Pipe

The Get-PvsServer output is piped to the Test-PvsDirectory.

```
Get-PvsServer -Name theServer -Fields Guid | Test-PvsDirectory -Path "C:\directory\subdirectory"
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

EXAMPLE 3: Test-PvsDirectory for Name with ReadOnly

```
Test-PvsDirectory -Name theServer -Path "C:\directory\subdirectory" -ReadOnly
```

EXAMPLE 4: Test-PvsDirectory for PvsServer Using Pipe

The Get-PvsServer output is piped to the Test-PvsDirectory.

```
Get-PvsServer -Name theServer -Fields Guid | Test-PvsDirectory -Path "C:\directory\subdirectory" -ReadOnly
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Unlock-PvsAllDisk

Remove all locks for a Disk.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to remove all locks for the Disk.

or this required & resolution

string[] Name or DiskLocatorName: Name of Disk Locator to remove all locks for the Disk.

One of these resolutions when needed

Guid[] SiteId: GUID of the Site.

string[] SiteName: Name of the Site.

One of these resolutions when needed

Guid[] StoreId: GUID of the Store that is needed when a DiskLocatorName is used.

string[] StoreName: Name of the Store that is needed when a DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

PvsObject[] Object: PvsObjects with the members below can be used as the Object parameter or from a pileline:

DiskLocatorId

Optional

SwitchParameter Confirm: The impact of this operation is "medium". If -Confirm is specified, the operation will be confirmed. $ConfirmPreference can be set to "medium" or "low" to have confirmation without the Confirm parameter.

EXAMPLE 1: Unlock-PvsAllDisk for Name

```
Unlock-PvsAllDisk -Name theDiskLocator -SiteName theSite -StoreName
theStore
```

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

EXAMPLE 2: Unlock-PvsAllDisk for PvsDiskLocator Using Pipe

The Get-PvsDiskLocator output is piped to the Unlock-PvsAllDisk.

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
theStore -Fields Guid | Unlock-PvsAllDisk
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Unlock-PvsDisk

Remove lock for the Disk.

This required

Guid[] Guid or DiskLocatorId: GUID of the Disk Locator to remove a lock for the Disk.

or this required & resolution

string[] Name or DiskLocatorName: Name of the Disk Locator to remove a lock for the Disk.

```
Optional

 Guid[] OwnerId: GUID of the Owner of the Disk Lock.

One of these resolutions when needed

 Guid[] SiteId: GUID of the Site.

 string[] SiteName: Name of the Site.

One of these resolutions when needed

 Guid[] StoreId: GUID of the Store that is needed when a
                 DiskLocatorName is used.

 string[] StoreName: Name of the Store that is needed when a
                 DiskLocatorName is used.

Instead of a parameter that matches one of the members listed

 PvsObject[] Object: PvsObjects with the members below can be used as
                 the Object parameter or from a pileline:

  DiskLocatorId

Optional

 SwitchParameter Confirm: The impact of this operation is "medium". If
                 -Confirm is specified, the operation will be
                 confirmed. $ConfirmPreference can be set to "medium"
                 or "low" to have confirmation without the Confirm
                 parameter.
```

EXAMPLE 1: Unlock-PvsDisk for Name

```
Unlock-PvsDisk -Name theDiskLocator -SiteName theSite -StoreName
                theStore

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 2: Unlock-PvsDisk for PvsDiskLocator Using Pipe

```
The Get-PvsDiskLocator output is piped to the Unlock-PvsDisk.

Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Unlock-PvsDisk

The -Fields parameter with only the needed fields specified makes the
                Get work faster because only those fields are
                retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 3: Unlock-PvsDisk for Name with OwnerId

```
Unlock-PvsDisk -Name theDiskLocator -SiteName theSite -StoreName
                theStore -OwnerId = "e6c9884e-067c-4924-a581-
                312c699b4bb6"

Guid can be used instead of Name so that the SiteName or SiteId and
                StoreName or StoreId are not also needed.
```

EXAMPLE 4: Unlock-PvsDisk for PvsDiskLocator Using Pipe

```
The Get-PvsDiskLocator output is piped to the Unlock-PvsDisk.
```

```
Get-PvsDiskLocator -Name theDiskLocator -SiteName theSite -StoreName
                theStore -Fields Guid | Unlock-PvsDisk -OwnerId =
                "e6c9884e-067c-4924-a581-312c699b4bb6"
```

The -Fields parameter with only the needed fields specified makes the Get work faster because only those fields are retrieved.

Guid can be used instead of Name so that the SiteName or SiteId and StoreName or StoreId are not also needed.

Update-PvsInventory

Force the Inventory service to refresh its Inventory Table.

EXAMPLE 1: Update-PvsInventory

```
Update-PvsInventory
```