Citrix Provisioning Services 2507 PowerShell

Contents

Citrix Provisioning 2507 PowerShell Programming Guide

January 13, 2026

The Citrix Provisioning PowerShell SDK automates complex and repetitive tasks. It provides the mechanism to set up and manage Citrix Provisioning environment.

NOTE:

You must have correct administrative privileges to use the programming commands. If you attempt to use the commands without having elevated privileges, the system displays an 'Invalid access' message.

Using the PowerShell Programmer Interface

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

Installation and use of PowerShell Module

The PowerShell snap-in (Citrix.PVS.SnapIn.dll) is installed as part of Provisioning Services Console install.

Once installed, you can load the powershell module into a powershell instance with the command:

```
1  Import-Module "path\Citrix.PVS.SnapIn.dll"
```

Where path is the installed location of the Provisioning Services console (default `C:\Program Files\Citrix\Provisioning Services Console`)

Short cmdlet list

January 13, 2026

- 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-PvsKeyRotationPendingServers
- Get-PvsLocalServer
- Get-PvsMaintenanceVersionExists
- Get-PvsMinimumLastAutoAddDeviceNumber
- Get-PvsMountedDisk
- Get-PvsMountedDriveLetter
- Get-PvsNewDiskVersion
- Get-PvsScrambledDataReport
- 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-PvsCreateBDM

- Start-PvsCreateDisk
- Start-PvsDeviceBoot
- Start-PvsDeviceDiskTempVersionMode
- Start-PvsDeviceReboot
- Start-PvsDeviceReconnect
- Start-PvsDeviceShutdown
- Start-PvsDeviceUpdateBdm
- Start-PvsDisplayMessage
- Start-PvsReportBug
- Start-PvsRotateEncryptionKeys
- Start-PvsStreamService
- Start-PvsUpdateTask
- Stop-PvsCreateDisk
- Stop-PvsDeviceDiskTempVersionMode
- Stop-PvsStreamService
- Stop-PvsTask
- Stop-PvsUpdateTask
- Test-PvsDirectory
- Test-PvsVirtualHostConnection
- Unlock-PvsAllDisk
- Unlock-PvsDisk
- Update-PvsInventory

Cmdlet by Object Type

January 13, 2026

Many

- Clear-PvsConnection
- Get-PvsAuditActionParameter
- Get-PvsAuditActionProperty
- Get-PvsAuditActionSibling
- Get-PvsAuditTrail
- Get-PvsConnection
- Get-PvsExists
- Get-PvsScrambledDataReport

- Reset-PvsDatabase
- Set-PvsConnection

AuthGroup

- Get-PvsAuthGroup
- Get-PvsAuthGroupUsage
- Grant-PvsAuthGroup
- New-PvsAuthGroup
- Remove-PvsAuthGroup
- Revoke-PvsAuthGroup
- Set-PvsAuthGroup

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-PvsCreateBDM
- Start-PvsDeviceBoot
- Start-PvsDeviceDiskTempVersionMode
- Start-PvsDeviceReboot
- Start-PvsDeviceReconnect
- 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
- Import-PvsDisk
- Invoke-PvsPromoteDiskVersion
- Invoke-PvsRevertDiskVersion
- Merge-PvsDisk
- Mount-PvsDisk
- New-PvsDiskMaintenanceVersion
- New-PvsDiskUpdateDevice
- Remove-PvsDiskFromUpdateTask
- Remove-PvsDiskUpdateDevice
- Remove-PvsDiskVersion
- Set-PvsDisk
- Set-PvsDiskUpdateDevice
- Set-PvsDiskVersion
- Set-PvsOverrideVersion
- 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
- Start-PvsRotateEncryptionKeys

FarmView

- Get-PvsFarmView
- New-PvsFarmView
- Remove-PvsFarmView
- Set-PvsFarmView

Server

- Copy-PvsServerProperties
- Get-PvsDirectory
- Get-PvsKeyRotationPendingServers
- 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

VirtualHostingPool

- Get-PvsVirtualHostingPool
- New-PvsVirtualHostingPool
- Remove-PvsVirtualHostingPool
- Set-PvsVirtualHostingPool
- Test-PvsVirtualHostConnection

CeipData

- Get-PvsCeipData
- Get-PvsUploadCeip
- New-PvsCeipData
- Set-PvsCeipData

CisData

- Get-PvsCisData
- New-PvsCisData
- Set-PvsCisData

Error codes (244 total):

January 13, 2026

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.

Code	Description
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.

	Code	Description
	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.

Code	Description
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 Citrix Provisioning server 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 Citrix Provisioning server 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.

	Code	Description
	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.

Code	Description
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].

Code	Description
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].

Code	Description
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].

Code	Description
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.

Code	Description
	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.

	Code	Description
		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.

	Code	Description
		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.

	Code	Description
	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.

	Code	Description
	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: Host connection cannot be deleted as provisioned devices are using it.
	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].

	Code	Description
	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 Citrix Provisioning 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.

	Code	Description
	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.

Code	Description
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.

	Code	Description
	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.

Code	Description
	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.

Code	Description
	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.

	Code	Description
		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.

Code	Description
	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.

	Code	Description
		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.

Code	Description
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.

	Code	Description
	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: Citrix Virtual Desktops Site for Devices is not valid, the Citrix Virtual Desktops Site is: [v1]
	215	XenDesktopServiceListOutOfDate: Citrix Virtual Desktops Site [v1] is not reachable, check that the Citrix PVS Soap Server service user has Citrix Virtual Desktops permissions and network connectivity.
	216	NoXenDesktopServiceForPersonalVDiskCapability: No Citrix Virtual Desktops service found for Personal vDisk capability.

	Code	Description
	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

	Code	Description
	232	CannotCreateRegKey: Cannot create Registry key [v1]
	233	DecryptionException: Cannot Decrypt registry value [v1].
	234	StkReadException: Cannot Read Citrix Cloud Information
	235	KeyCyclingOnGoingException: A Key Rotation Operation is ongoing, please retry later
	236	VirtualHostConnectionError: The Virtual Host Connection could not be established.
	237	RotateCvadSiteKeyError: Cannot Rotate Key as no public key found for the specified sites.
	238	ControllerConnectionError: The Citrix Virtual Desktops controller at [v1] could not be reached.
	239	ControllerAuthorizationError: Access to the Citrix Virtual Desktops controller at [v1] is denied.
	240	SuspendedDevice: The task cannot be performed on suspended or paused Devices. Shut down the Devices before attempting to perform the task.
	241	DeviceStateUnknown: The Device state is unknown.
	242	ControllerPermissionsError: The Current user needs to be a Machine Catalog Administrator or higher on the Citrix Virtual Desktops controller at [v1]. [v2][v3].
	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].

Object types in the Citrix.PVS.SnapIn Namespace:

January 13, 2026

PvsADAccount

Read-Only Fields

Field	Type	Description
Domain	string	Domain the account is a member of.
DomainController	string	The name of the DC used to create the host's computer account.
Name	string	Name of the Device for the account.
Sid	string	The value of the objectSID AD attribute of the same name for the Device's computer account.

PvsAuditAction

Read-Only Fields

Field	Type	Description
Guid or AuditActionId	Guid	GUID of the action.
ObjectId	Guid	GUID of the object of the action.
ObjectName	string	Name of the object of the action. Max Length=1000
Path	string	Path of the object of the action. An example is Site\Collection for a Device. Default='"'Max Length=101
SiteId	Guid	GUID of the Site for the object of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000

Field	Type	Description
SubId	Guid	GUID of the Collection or Store of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000
Type	uint	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

Field	Type	Description
AuditActionId	Guid	GUID of the Audit Action used for Get and Set.
Name or AuditParameterName	string	Name of the parameter. Max Length=50
Value	string	Value of the parameter. Max Length=1000

PvsAuditActionProperty

Read-Only Fields

Field	Type	Description
AuditActionId	Guid	GUID of the Audit Action used for Get and Set.
Name or AuditPropertyName	string	Name of the property. Max Length=50
NewValue	string	New value of the Property. Default=’”’Max Length=1000
OldValue	string	Previous value of the Property. Default=’”’Max Length=1000

PvsAuditTrail

Read-Only Fields

Field	Type	Description

Field	Type	Description
Action	uint	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),

Field	Type	Description
Attachments	uint	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 or AuditActionId	Guid	GUID of the action.
Domain	string	Domain of the user that performed the action. Max Length=255
ObjectId	Guid	GUID of the object of the action. Default=00000000-0000-0000-0000-000000000000
ObjectName	string	Name of the object of the action. Default='"'Max Length=1000
ParentId	Guid	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
Path	string	Path of the object of the action. An example is Site\Collection for a Device. Default='"'Max Length=101

Field	Type	Description
RootId	Guid	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
SiteId	Guid	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	GUID of the Collection or Store of the action. 00000000-0000-0000-0000-000000000000 when not valid. Default=00000000-0000-0000-0000-000000000000
Time	DateTime	Date/Time the action occurred down to the millisecond. Has the date and time including milliseconds. Default=Empty
Type	uint	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)

Field	Type	Description
UserName	string	User that performed the action. Max Length=255

PvsAuthGroup

Read/Write Fields

Field	Type	Description
Name or AuthGroupName	string	Name of the Active Directory or Windows Group. Max Length=450
Description	string	User description. Default=”” Max Length=250
IsReadOnly	bool	Indicates whether the administrative role is read-only(true) or read-write(false). Default=false

Read-Only Fields

Field	Type	Description
Guid or AuthGroupId	Guid	Read-only GUID that uniquely identifies this AuthGroup.
Role	uint	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

Field	Type	Description
Guid or Id	Guid	GUID of the item. The item can be a Farm, Site or Collection. It will be 00000000-0000-0000-0000-000000000000 for Farm.
Name	string	Name of the item. The item can be a Farm, Site or Collection.
Role	uint	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

Field	Type	Description
Enabled	uint	1 if CEIP is enabled, otherwise 0. Min=0, Max=1
InProgress	uint	1 if an upload is currently in progress, otherwise 0. Default=0
NextUpload	DateTime	Date and time next CEIP upload is due if enabled is 1. Default=Empty
OneTimeUpload	uint	1 to perform a one time upload. Default=0
ServerId	Guid	ID of server that is currently uploading, null if InProgress is 0. Default=00000000-0000-0000-0000-000000000000

Read-Only Field

Field	Type	Description
Uuid	Guid	CEIP UUID.

PvsCisData

Read/Write Fields

Field	Type	Description
Password	string	Password of the user required to obtain the token. This is required only by Set and Add
Path	string	Path where the last problem report bundle was saved Default=’”’Max Length=255
UserName	string	Username used to obtain the token Default=’”’Max Length=255

Read-Only Fields

Field	Type	Description
Guid or CisDataId	Guid	CIS UUID
UploadToken	string	Token for uploading bundles to CIS Default=’”’Max Length=65535

PvsCollection

Read/Write Fields

Field	Type	Description
AutoAddNumberLength	uint	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 ((lenautoAddPre-fix+lenautoAddSuffix)+AutoAddNumberLeng Min=3, Max=9, Default=4
AutoAddPrefix	string	The string put before the Device Number for Auto Add. Default=’”’ASCII computer name characters no end digit Max Length=12
AutoAddSuffix	string	The string put after the Device Number for Auto Add. Default=’”’ASCII computer name characters no begin digit Max Length=12
AutoAddZeroFill	bool	True when zeros be placed before the Device Number up to the AutoAddNumberLength for Auto Add, false otherwise. Default=true
Name or CollectionName	string	Name of the Collection. It is unique within the Site. Max Length=50
Description	string	User description. Default=’”’ Max Length=250
Enabled	bool	True when Devices in the Collection can be booted, false otherwise. Default=true
LastAutoAddDeviceNumber	uint	The Device Number of the last Auto Added Device. Default=0

Field	Type	Description
ProvisioningType	uint	The provisioning facility which created the Devices in this Collection. 0 is PVS, 1 is Studio. Only writable when adding a colection. Min=0, Max=1, Default=0
TemplateDeviceId	Guid	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	string	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

Field	Type	Description
ActiveDeviceCount	uint	Read-only count of active Devices in this Collection. Default=0
Guid or CollectionId	Guid	Read-only GUID that uniquely identifies this Collection.
DeviceCount	uint	Read-only count of Devices in this Collection. Default=0
DeviceWithPVDCount	uint	Read-only count of Devices with Personal vDisk in this Collection. Default=0
MakActivateNeededCount	uint	Read-only count of active Devices that need MAK activation in this Collection. Default=0

Field	Type	Description
Role	uint	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
SiteId	Guid	GUID of the Site that this Collection is a member of. It is not used with SiteName.
SiteName	string	Name of the Site that this Collection is a member of. It is not used with SiteId.

PvsConnection

Read/Write Fields

Field	Type	Description
Domain	string	Domain name to use for Authentication. If it has a value, it will be *****. Default=’”’
Password	string	Password to use for Authentication. If it has a value, it will be *****. Default=’”’
Persist	string	True when the connection settings should be, for Set, or have been, for Get, saved to the registry.
Port	string	The Port to use to connect. Default=54321
Name or Server	string	Name or IP of the Server to connect to. Default=localhost
User	string	User name to use for Authentication. If it has a value, it will be *****. Default=’”’

Read-Only Field

Field	Type	Description
Connected	string	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

Field	Type	Description
AdPassword	string	The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=””ASCII Max Length=65535
AdSignature	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0

Field	Type	Description
AdTimestamp	uint	The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0
Authentication	uint	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
BdmBoot	bool	Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false
BdmCreated	DateTime	Timestamp when BDM device was created Default=Empty
BdmFormat	uint	1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0
BdmType	uint	Use PXE boot when set to 0, BDM (Bios) boot when set to 1 and BDM (Uefi) boot when set to 2. Default=0
BdmUpdated	DateTime	Timestamp of the last BDM boot disk update. Default=Empty
BootFrom	uint	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	string	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

Field	Type	Description
Description	string	User description. Default=’”’ Max Length=250
DeviceMac	PvsPhysicalAddress	Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device. Default=’”’
Name or DeviceName	string	Computer name with no spaces. ASCII computer name characters Max Length=15
DomainControllerName	string	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
DomainName	string	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	string	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
DomainTimeCreated	DateTime	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
EkPub	string	Client TPM EK public key Default=’”’
Enabled	bool	True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true

Field	Type	Description
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0
LocalDiskEnabled	bool	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
LocalWriteCacheDiskSize	uint	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
LogLevel	uint	Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
Password	string	Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default=””ASCII Max Length=65535
Port	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901
ProvisioningType	uint	The provisioning facility which created the device. 0 is PVS, 1 is Studio. Only writable when adding a device. Min=0, Max=1, Default=0

Field	Type	Description
SecurityPolicy	uint	Client security policy Expected values are 0, 2 or 10, Default=null.
Type	uint	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
UpdatePageFileSettings	bool	Update pagefile settings Default=false
User	string	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
WriteCacheDisk	string	Write cache disk number Default=’”’Max Length=3
WriteCachePartition	string	Write cache disk partition number Default=’”’Max Length=3
XsPvsProxyUuid	Guid	UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000

Read-Only Fields

Field	Type	Description
Active	bool	True if the Device is currently active, false otherwise. Default=false
CollectionId	Guid	GUID of the Collection this Device is to be a member of. It is not used with CollectionName.
CollectionName	string	Name of the Collection this Device is to be a member of. SiteName or SiteId must also be used.
Guid or DeviceId	Guid	Read-only GUID that uniquely identifies this Device.
HypVmId	string	Hypervisor VM ID for HCL Default=’”’Max Length=65535
PvdDriveLetter	string	Read-only Personal vDisk Drive letter. Range is E to U and W to Z. Default=’”’Max Length=1
Role	uint	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
SiteId	Guid	GUID of the Site the CollectionName is to be a member of. This or SiteName is used with CollectionName.
SiteName	string	Name of the Site the CollectionName is to be a member of. This or SiteId is used with CollectionName.
Template	bool	True if the Device is the template in its Collection, false otherwise. Default=false

Field	Type	Description
TemporaryVersionSet	bool	Read-only true when temporary version is set. Default=false
VirtualHostingPoolId	Guid	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

Name	Description
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.

- 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

Field	Type	Description
Name or Bootstrap	string	Name of the bootstrap file. Max Length=259
MenuText	string	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

Field	Type	Description
Guid or DeviceId	Guid	Read-only GUID that uniquely identifies the Device with temporary version.
Name or DeviceName	string	Read-only Computer name that uniquely identifies the Device with temporary version. ASCII computer name characters
DiskLocatorId	Guid	Read-only GUID that uniquely identifies then Disk Locator with temporary version.
DiskLocatorName	string	Read-only Name of the Disk Locator File with temporary version. It is unique within the Store. ASCII
SiteId	Guid	Read-only GUID of the Site the Device and DiskLocator are a member of.
SiteName	string	Read-only Name of the Site the Device and DiskLocator are a member of.

Field	Type	Description
StoreId	Guid	Read-only GUID of the Store that the Disk Locator is a member of.
StoreName	string	Read-only Name of the Store that the Disk Locator is a member of.
Version	uint	Read-only Disk version the temporary is for.

PvsDeviceInfo

Read-Only Fields

Field	Type	Description
Active	bool	True if the Device is currently active, false otherwise. Default=false
AdPassword	string	The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=”” ASCII Max Length=65535
AdSignature	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0
AdTimestamp	uint	The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

Field	Type	Description
Authentication	uint	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
BdmBoot	bool	Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false
BdmCreated	DateTime	Timestamp when BDM device was created Default=Empty
BdmFormat	uint	1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0
BdmType	uint	Use PXE boot when set to 0, BDM (Bios) boot when set to 1 and BDM (Uefi) boot when set to 2. Default=0
BdmUpdated	DateTime	Timestamp of the last BDM boot disk update. Default=Empty
BootFrom	uint	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	string	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
CollectionId	Guid	GUID of the Collection this Device is to be a member of. It is not used with CollectionName.

Field	Type	Description
CollectionName	string	Name of the Collection this Device is to be a member of. SiteName or SiteId must also be used.
Description	string	User description. Default=”” Max Length=250
Guid or DeviceId	Guid	Read-only GUID that uniquely identifies this Device.
DeviceMac	PvsPhysicalAddress	Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device. Default=””
Name or DeviceName	string	Computer name with no spaces. ASCII computer name characters Max Length=15
DiskFileName	string	Name of the Disk File including the extension. It is equal to “”if the Device is not active.
DiskLocatorId	Guid	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	string	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	uint	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

Field	Type	Description
DiskVersionAccess	uint	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
DomainControllerName	string	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
DomainName	string	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	string	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
DomainTimeCreated	DateTime	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
EkPub	string	Client TPM EK public key Default=''

Field	Type	Description
Enabled	bool	True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0
HypVmId	string	Hypervisor VM ID for HCL Default=”” Max Length=65535
Ip	System.Net.IPAddress	Read-only IPv4 Address of the Device. It is equal to 0.0.0.0 if the Device is not active or not using IPv4
Ipv6	string	Read-only IPv6 Address of the device. It is equal to “::” if the device is not active or not using IPv6 Default=0
License	uint	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
LicenseType	uint	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
LocalDiskEnabled	bool	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
LocalWriteCacheDiskSize	uint	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

Field	Type	Description
LogLevel	uint	Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
MakLicenseActivated	uint	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	string	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.
Password	string	Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default=’”’ASCII Max Length=65535
Port	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901
ProvisioningType	uint	The provisioning facility which created the device. 0 is PVS, 1 is Studio. Only writable when adding a device. Min=0, Max=1, Default=0

Field	Type	Description
PvdDriveLetter	string	Read-only Personal vDisk Drive letter. Range is E to U and W to Z. Default=’”’Max Length=1
Role	uint	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
SecurityPolicy	uint	Client security policy Expected values are 0, 2 or 10, Default=null.
ServerId	Guid	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.
ServerIpConnection	System.Net.IPAddress	Read-only IPv4 Address of the Server that the Device is using. It is equal to 0.0.0.0 if the Device is not active or not using IPv4.
ServerIpv6Connection	string	Read-only IPv6 Address of the Server that the Device is using. It is equal to “::”if the Device is not active or not using IPv6. Default=0
ServerName	string	Read-only Name of the Server that the Device is using. It is equal to “”if the Device is not active.
ServerPortConnection	uint	Read-only Port of the Server that the Device is using. It is equal to 0 if the Device is not active. Default=0

Field	Type	Description
SiteId	Guid	GUID of the Site the CollectionName is to be a member of. This or SiteName is used with CollectionName.
SiteName	string	Name of the Site the CollectionName is to be a member of. This or SiteId is used with CollectionName.
Status	string	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.
Template	bool	True if the Device is the template in its Collection, false otherwise. Default=false
TemporaryVersionSet	bool	Read-only true when temporary version is set. Default=false
Type	uint	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
UpdatePageFileSettings	bool	Update pagefile settings Default=false
User	string	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

Field	Type	Description
VirtualHostingPoolId	Guid	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
WriteCacheDisk	string	Write cache disk number Default=””Max Length=3
WriteCachePartition	string	Write cache disk partition number Default=””Max Length=3
XsPvsProxyUuid	Guid	UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000

PvsDevicePersonality

Read-Only Fields

Name	Description
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.

- 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

Field	Type	Description
Name	string	Name of the Device personality item. Max Length=250
Value	string	Value for the Device personality item. Max Length=1000

PvsDeviceStatus

Read-Only Fields

Field	Type	Description
Guid or DeviceId	Guid	Read-only GUID of the Device. Can be used with Get Device.
Name or DeviceName	string	Read-only Name of the Device. Can be used with Get Device.
DiskFileName	string	Name of the Disk File including the extension.
DiskLocatorId	Guid	Read-only GUID of the Disk Locator that the Device is using.
DiskLocatorName	string	Read-only name of the Disk Locator File that the Device is using.
DiskVersion	uint	Read-only version of the Disk Locator File that the Device is using. Default=0

Field	Type	Description
DiskVersionAccess	uint	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
Ip	System.Net.IPAddress	Read-only IPv4 Address of the Device.
Ipv6	string	Read-only IPv6 Address of the device Default=0
LicenseType	uint	0 when None, 1 for Desktop, 2 for Server, 5 for OEM SmartClient, 6 for XenApp, 7 for XenDesktop. Default=0
MakLicenseActivated	uint	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
QuicCipherSuite	uint	Read-only The QUIC cipher suite that the Device is using for streaming. It is none if legacy protocol is used. Default=””
ServerId	Guid	Read-only GUID of the Server that the Device is using.
ServerIpConnection	System.Net.IPAddress	Read-only IPv4 Address of the Server that the Device is using.
ServerIpv6Connection	string	Read-only IPv6 Address of the Server that the Device is using. Default=0

Field	Type	Description
ServerName	string	Read-only Name of the Server that the Device is using.
ServerPortConnection	uint	Read-only Port of the Server that the Device is using. Default=0
Status	string	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.
StreamProtocol	uint	Read-only The stream protocol that the Device is using: 0(Legacy protocol), 1(QUIC protocol). Min=0, Max=1, Default=0

PvsDisk

Read/Write Fields

Field	Type	Description
AccelerateOfficeActivation	bool	Run script to activate office automatically.
ActivationDateEnabled	bool	Use activation date to activate image when set to true. Default false
ActiveDate	DateTime	Date to activate the disk if AutoUpdateEnabled and activationDateEnabled are true. Has the date. Empty when the AutoUpdateEnabled or activationDateEnabled are false.
AdPasswordEnabled	bool	Enable AD password management when set to true.
Author	string	User defined author. Max Length=40

Field	Type	Description
AutoUpdateEnabled	bool	Automatically update this image for matching Devices when set to true. Default false
Build	UInt64	User defined build number. Min=0, Max=4294967295, Default=0
Class	string	Class of the Disk. Max Length=40
ClearCacheDisabled	string	Clear cached secrets disabled.
Company	string	User defined company. Max Length=40
Date	string	User defined date. Max Length=40
EnableSetLocalTime	string	Enable setting the local time bias on the disk version.
VHDX	bool	If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false
HaEnabled	bool	Enable HA when set to true.
HardwareTarget	string	User defined hardware target. Max Length=127
ImageType	string	Type of this image (software type). Max Length=40
InternalName	string	User defined name. Max Length=63
LicenseMode	uint	0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0
LongDescription	string	Description of the Disk. Max Length=399
MajorRelease	UInt64	User defined major release number. Min=0, Max=4294967295, Default=0
MinorRelease	UInt64	User defined minor release number. Min=0, Max=4294967295, Default=0

Field	Type	Description
OperatingSystem	string	Operating System of Disk. Max Length=250
OriginalFile	string	User defined original file. Max Length=127
OsType	string	Operating System Type of Disk. Max Length=40
PrinterManagementEnabled	bool	Invalid printers will be deleted from the Device when set to true.
SerialNumber	string	User defined serial number. Max Length=36
Title	string	User defined title. Max Length=40
WriteCacheSize	UInt64	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
WriteCacheType	uint	0 (Private), 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), 9 (Cache in Device RAM with Overflow on Hard Disk), 10 (Private async), 11(Server persistent async), 12 (Cache in Device RAM with Overflow on Hard Disk async) Min=0, Max=12, Default=0

Read-Only Fields

Field	Type	Description
Guid or DiskLocatorId	Guid	GUID of the DiskLocator used for Get and Set.
DiskSize	UInt64	Read-only size of the image. The value is 0 when it is not available. Default=0
LogicalSectorSize	string	Logical Sector Size. Values are: 512, 4096, Default=512
VhdBlockSize	uint	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

Field	Type	Description
AccelerateOfficeActivation	bool	Run script to activate office automatically.
ActivationDateEnabled	bool	Use activation date to activate image when set to true. Default false
Active	bool	True if the DiskLocator is currently active, false otherwise. Default=false

Field	Type	Description
ActiveDate	DateTime	Date to activate the disk if AutoUpdateEnabled and activationDateEnabled are true. Has the date. Empty when the AutoUpdateEnabled or activationDateEnabled are false.
AdPasswordEnabled	bool	Enable AD password management when set to true.
Author	string	User defined author. Max Length=40
AutoUpdateEnabled	bool	Automatically update this image for matching Devices when set to true. Default false
Build	UInt64	User defined build number. Min=0, Max=4294967295, Default=0
Class	string	Class of the Disk. Max Length=40
ClearCacheDisabled	string	Clear cached secrets disabled.
Company	string	User defined company. Max Length=40
Date	string	User defined date. Max Length=40
Description	string	User description. Default=’”’ Max Length=250
DeviceCount	uint	Read-only count of Devices. Default=0
Guid or DiskLocatorId	Guid	Read-only GUID that uniquely identifies this Disk Locator.
Name or DiskLocatorName	string	Name of the Disk Locator File. It is unique within the Store. ASCII Max Length=52
DiskSize	UInt64	Read-only size of the image. The value is 0 when it is not available. Default=0

Field	Type	Description
DiskUpdateDeviceId	Guid	GUID of the DiskUpdateDevice that is used when updates are performed. Default=00000000-0000-0000-0000-000000000000
DiskUpdateDeviceName	string	Name of the DiskUpdateDevice that is used when updates are performed. Default=’”’
Enabled	bool	True when this disk can be booted, false otherwise. Default=true
EnabledForDevice	bool	True when this disk is enabled for the Device specified, false otherwise. This is only returned when a Device is specified. Default=true
EnableSetLocalTime	string	Enable setting the local time bias on the disk version.
VHDX	bool	If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false
HaEnabled	bool	Enable HA when set to true.
HardwareTarget	string	User defined hardware target. Max Length=127
ImageType	string	Type of this image (software type). Max Length=40
InternalName	string	User defined name. Max Length=63
LicenseMode	uint	0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0
Locked	bool	True if the Disk is currently locked, false otherwise. Default=false

Field	Type	Description
LogicalSectorSize	string	Logical Sector Size. Values are: 512, 4096, Default=512
LongDescription	string	Description of the Disk. Max Length=399
MajorRelease	UInt64	User defined major release number. Min=0, Max=4294967295, Default=0
Mapped	bool	True if the Disk is currently mapped, false otherwise. Default=false
MenuText	string	Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default=’”’ASCII Max Length=64
MinorRelease	UInt64	User defined minor release number. Min=0, Max=4294967295, Default=0
OperatingSystem	string	Operating System of Disk. Max Length=250
OriginalFile	string	User defined original file. Max Length=127
OsType	string	Operating System Type of Disk. Max Length=40
PrinterManagementEnabled	bool	Invalid printers will be deleted from the Device when set to true.
RebalanceEnabled	bool	True when this Server can automatically rebalance Devices, false otherwise. Default=false
RebalanceTriggerPercent	uint	Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25

Field	Type	Description
Role	uint	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
SerialNumber	string	User defined serial number. Max Length=36
ServerId	Guid	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	string	Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=’”’
SiteId	Guid	GUID of the Site this DiskLocator is to be a member of. It is not used with SiteName.
SiteName	string	Name of the Site this DiskLocator is to be a member of. It is not used with SiteId.
StoreId	Guid	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	string	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.
SubnetAffinity	uint	Qualifier for subnet affinity when assigning a Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2, Default=0

Field	Type	Description
TemporaryVersionSet	bool	Read-only true when temporary version(s) are set. Default=false
Title	string	User defined title. Max Length=40
VhdBlockSize	uint	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
WriteCacheSize	UInt64	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
WriteCacheType	uint	0 (Private), 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), 9 (Cache in Device RAM with Overflow on Hard Disk), 10 (Private async), 11(Server persistent async), 12 (Cache in Device RAM with Overflow on Hard Disk async) Min=0, Max=12, Default=0

PvsDiskInventory

Read-Only Fields

Field	Type	Description
Active	string	1 if the Server is currently active, 2 if unknown, and 0 otherwise.
Guid or DiskLocatorId	Guid	GUID of the DiskLocator used for Get and Set.
FilePath	string	Path used to access the disk version from the Server. Empty if the information is not available.
FileTime	DateTime	Date/Time of the date version file. Has the date and time without milliseconds. Empty if the information is not available.
PropertiesTime	DateTime	Date/Time of the disk properties. Has the date and time without milliseconds. Empty if the information is not available.
ServerId	Guid	GUID of the Server that the Disk Version Inventory is being reported about.
ServerName	string	Name of the Server that the Disk Version Inventory is being reported about.
State	string	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).

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

PvsDiskLocator

Read/Write Fields

Field	Type	Description
Description	string	User description. Default=”” Max Length=250
Enabled	bool	True when this disk can be booted, false otherwise. Default=true
MenuText	string	Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default=””ASCII Max Length=64
RebalanceEnabled	bool	True when this Server can automatically rebalance Devices, false otherwise. Default=false
RebalanceTriggerPercent	uint	Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25
ServerId	Guid	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	string	Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=””

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

Read-Only Fields

Field	Type	Description
Active	bool	True if the DiskLocator is currently active, false otherwise. Default=false
Guid or DiskLocatorId	Guid	Read-only GUID that uniquely identifies this Disk Locator.
Name or DiskLocatorName	string	Name of the Disk Locator File. It is unique within the Store. ASCII Max Length=52
DiskUpdateDeviceId	Guid	GUID of the DiskUpdateDevice that is used when updates are performed. Default=00000000-0000-0000-0000-000000000000
DiskUpdateDeviceName	string	Name of the DiskUpdateDevice that is used when updates are performed. Default=””
EnabledForDevice	bool	True when this disk is enabled for the Device specified, false otherwise. This is only returned when a Device is specified. Default=true
Mapped	bool	True if the Disk is currently mapped, false otherwise. Default=false
Role	uint	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

Field	Type	Description
SiteId	Guid	GUID of the Site this DiskLocator is to be a member of. It is not used with SiteName.
SiteName	string	Name of the Site this DiskLocator is to be a member of. It is not used with SiteId.
StoreId	Guid	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	string	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.
TemporaryVersionSet	bool	Read-only true when temporary version(s) are set. Default=false

PvsDiskLocatorLock

Read-Only Fields

Field	Type	Description
DeviceId	Guid	GUID of the Device that has the lock, will be 00000000-0000-0000-0000-000000000000 if a Server has the lock.
DeviceName	string	Name of the Device that has the lock, will not be included if a Server has the lock.
Exclusive	bool	True when the lock is exclusive, false when it is shared. Default=false

Field	Type	Description
ReadOnly	bool	True when lock is because file system is read only, false when file system is read write Default=false
ServerId	Guid	GUID of the Server that has the lock, will be 00000000-0000-0000-0000-000000000000 if a Device has the lock.
ServerName	string	Name of the Server that has the lock, will not be included if a Device has the lock.

PvsDiskUpdateDevice

Read/Write Fields

Field	Type	Description
AdPassword	string	The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=’”’Max Length=65535
AdSignature	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0
AdTimestamp	uint	The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0
Description	string	User description. Default=’”’ Max Length=250

Field	Type	Description
DomainControllerName	string	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
DomainName	string	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	string	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
DomainTimeCreated	DateTime	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
LogLevel	uint	Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
Port	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901

Read-Only Fields

Field	Type	Description
Active	bool	True if the Device is currently active, false otherwise. Default=false
Guid or DeviceId	Guid	Read-only GUID that uniquely identifies this Device.
DeviceMac	PvsPhysicalAddress	Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device. Default=””
Name or DeviceName	string	Computer name with no spaces. ASCII computer name characters Max Length=15
DiskLocatorId	Guid	GUID of the Disk Locator to update with this Device.
DiskLocatorName	string	Name of the Disk Locator File to update with this Device.
DiskVersion	uint	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
Ip	System.Net.IPAddress	Read-only IPv4 Address of the Device. It is equal to 0.0.0.0 if the Device is not active or not using IPv4
Ipv6	string	Read-only IPv6 Address of the device. It is equal to “::”if the device is not active or not using IPv6 Default=0
License	uint	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

Field	Type	Description
LicenseType	uint	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	uint	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	string	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.
ServerId	Guid	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.
ServerIpConnection	System.Net.IPAddress	Read-only IPv4 Address of the Server that the Device is using. It is equal to 0.0.0.0 if the Device is not active or not using IPv4.
ServerIpv6Connection	string	Read-only IPv6 Address of the Server that the Device is using. It is equal to “::”if the Device is not active or not using IPv6. Default=0
ServerName	string	Read-only Name of the Server that the Device is using. It is equal to “”if the Device is not active.

Field	Type	Description
ServerPortConnection	uint	Read-only Port of the Server that the Device is using. It is equal to 0 if the Device is not active. Default=0
SiteId	Guid	GUID of the Site this Disk Update Device is to be a member of.
SiteName	string	Name of the Site this Disk Update Device is to be a member of.
Status	string	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.
StoreId	Guid	GUID of the Store that the Disk Locator is a member of.
StoreName	string	Name of the Store that the Disk Locator is a member of.
VirtualHostingPoolId	Guid	GUID of the Virtual Hosting Pool. It is not used with VirtualHostingPoolName. Default=00000000-0000-0000-0000-000000000000
VirtualHostingPoolName	string	Name of the Virtual Hosting Pool.

PvsDiskUpdateStatus

Read-Only Fields

Field	Type	Description
CurrentStatus	uint	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	string	Message string that includes the results of the run. Default=””Max Length=255
Description	string	User description of the Update Task.
DeviceId	Guid	GUID that Device being used to do the update.
DeviceName	string	Name of the Device being used to do the update.
DiskLocatorId	Guid	GUID of the Disk Locator to update.
Name or DiskLocatorName	string	Name of the Disk Locator File to update.
Guid or DiskUpdateTaskId	Guid	GUID that uniquely identifies this Update Task and Device relationship.

Field	Type	Description
PreviousResult	uint	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	string	Message string that includes the results of the last run. Default=””Max Length=255
SiteId	Guid	GUID of the Site that this Update Task Name is a member of.
SiteName	string	Name of the Site that this Update Task Name is a member of.
StoreId	Guid	GUID of the Store that the Disk Locator is a member of.
StoreName	string	Name of the Store that the Disk Locator is a member of.
UpdateTaskId	Guid	GUID that uniquely identifies the Update Task.
UpdateTaskName	string	Name of the Update Task.
VirtualHostingPoolId	Guid	GUID of the Virtual Hosting Pool being used for the update.
VirtualHostingPoolName	string	Name of the Virtual Hosting Pool being used for the update.

PvsDiskVersion

Read/Write Fields

Field	Type	Description
Description	string	User description. Default=”” Max Length=250
ScheduledDate	DateTime	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

Field	Type	Description
Access	uint	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
CanDelete	bool	Read-only true when the version can be deleted. Default=false
CanMerge	bool	Read-only true when the version can be update merged. Will be set for the highest version number. Default=false

Field	Type	Description
CanMergeBase	bool	Read-only true when the version can be base merged. Will be set for the highest version number. Default=false
CanOverride	bool	Read-only true when the version can be set as the Override. Default=false
CanPromote	bool	Read-only true when the version can be promoted. Default=false
CanRevertMaintenance	bool	Read-only true when the version can be reverted to Maintenance Access. Default=false
CanRevertTest	bool	Read-only true when the version can be reverted to Test Access. Default=false
CanSetScheduledDate	bool	Read-only true when the version can have the scheduled date modified. Default=false
CreateDate	string	Read-only Date/Time that the Disk Version was created. Default=getdate
DeleteWhenFree	bool	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
DeviceCount	uint	Read-only count of Devices. Default=0
Name or DiskFileName	string	Name of the Disk File including the extension. Default=””
Guid or DiskLocatorId	Guid	GUID of the DiskLocator used for Get and Set.
GoodInventoryStatus	bool	True when the up to date file is accessible by all Servers, false otherwise. Default=false

Field	Type	Description
IsPending	bool	Read-only true when the version ScheduledDate has not occurred. Default=false
LocalTimeBias	string	The time zone set in the vDisk version as the offset from UTC in minutes. -1 indicates vDisk timezone is not known or applicable. Default is -1 Default=-1 Max Length=5
TaskId	uint	When a Merge is occurring, this will be set with the task number of the process that is occurring. Default=””
TemporaryVersionSet	bool	Read-only true when temporary version(s) are set. Some changes cannot be made to the version when this is set. Default=false
Type	uint	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
Version	uint	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

Field	Type	Description
AuditingEnabled	bool	True when Auditing is enabled, false otherwise. Default=false
AutoAddEnabled	bool	True when Auto Add is enabled, false otherwise. Default=false
AutomaticMergeEnabled	bool	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
DefaultDeviceLogLevel	uint	The default logging level used by newly created target devices. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
DefaultSiteId	Guid	GUID of the Site to place new Devices into automatically. Not used with defaultSiteName. Default=00000000-0000-0000-0000-000000000000
DefaultSiteName	string	Name of the Site to place new Devices into automatically. Not used with DefaultSiteId. Default=’”’
Description	string	User description. Default=’”’ Max Length=250
EncryptionStatus	uint	The current status for database encryption. 0 = Idle, 1 = Distributing Keys, 2= Re-encrypting database Min=0, Max=2, Default=0

Field	Type	Description
FarmEpoch	UInt64	Timestamp when the Data Encryption Key (DEK) was created and stored in the Credential Wallet. Default=0
Name or FarmName	string	Name of the Farm. Default=’”’ Max Length=50
IsUsageDataScrambled	bool	Indicates whether the sensitive information needs to be scrambled (true) or not (false) when uploading usage telemetry data. Default = true. Default=true
LastAuditArchiveDate	DateTime	Last date of Audit Trail data that was Archived. Has the date. Default=Empty
LicenseServer	string	License server name. Default=’”’Max Length=255
LicenseServerCertThumbprint	string	Stores the license server’s server certificate thumbprint when the server certificate is explicitly accepted by the administrator. NULL if no certificate is explicitly accepted. Default=’”’Max Length=65535
LicenseServerPort	uint	License server port. Min=1025, Max=65534, Default=27000
LicenseSKU	uint	LicenseSKU. 0 for on-premises, 1 for cloud. Min=0, Max=1, Default=0
LicenseTradeUp	bool	License server trade up, when set to true. Default=false
LicenseWebServicesPort	uint	The license server web services for licensing port. Min=1025, Max=65534, Default=8083
MaxVersions	uint	Maximum number a versions of a Disk that can exist before a merge will automatically occur. Min=3, Max=50, Default=5

Field	Type	Description
MergeMode	uint	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
OfflineDatabaseSupportEnabled	bool	True when Offline Database Support is enabled, false otherwise. Default=false

Read-Only Fields

Field	Type	Description
AdGroupsEnabled	bool	Active Directory groups are used for authorization, when set to true. Windows groups are used when set to false. Default=false
CloudSetupActive	bool	True if farm is integrated with Citrix Cloud and all PVS servers have been registered with Citrix Cloud. Default=false
CustomerId	string	The Citrix Cloud customer ID if the SetupType is Citrix Cloud. Default=’”’Max Length=128
CustomerName	string	The Citrix Cloud customer name if the SetupType is Citrix Cloud. Default=’”’Max Length=256
DatabaseInstanceName	string	Read-only name of the database instance.
DatabaseName	string	Read-only name of the database.
DatabaseServerName	string	Read-only name of the database server.

Field	Type	Description
EntitledState	string	Indicates the entitled state of PVS Farm with LAS. 0 (not entitled), 1 (entitled via LAS), 2 (entitled via V6)
EntitlementExpirationDate	string	Expiration date of the entitlement
FailoverPartnerInstanceName	string	Read-only name of the database server instance.
FailoverPartnerServerName	string	Read-only name of the database server.
Guid or FarmId	Guid	Read-only GUID that uniquely identifies this Farm.
IsReadOnly	string	Indicates whether the administrative role is read-only(true) or read-write(false).
KeyExchangeType	uint	The encryption key exchange type. 0=CredentialWallet, 1=Peer-2-Peer. Min=0, Max=1, Default=1
KeyRotationInitiatingServerFqdn	string	The FQDN of the Server who initiated the database encryption key rotation. Default=’”’Max Length=1024
ManualKeyCyclingRequired	bool	Indicates if a manual encryption key rotation is required(true) or not(false). Default=false
MultiSubnetFailover	string	Read-only Database MultiSubnetFailover value
Role	uint	Read-only Role of the user for this item. 100 is Farm Administrator, and 999 is read-only. Default=999

Field	Type	Description
SetupType	uint	Farm integration type. 0 for no integration, 1 for Citrix Cloud, 2 for Citrix Virtual Apps and Desktops. Min=0, Max=2, Default=0
XdSetupActive	bool	True if farm is integrated with Citrix Virtual Apps and Desktops and all PVS servers have been registered with the CVAD site. Default=false
XdSiteId	Guid	The CVAD site ID if the SetupType is Citrix Virtual Apps and Desktops. Default=00000000-0000-0000-0000-000000000000
XdSiteName	string	The CVAD site name if the SetupType is Citrix Virtual Apps and Desktops. Default=’”’Max Length=256

PvsFarmView

Read/Write Fields

Field	Type	Description
Description	string	User description. Default=’”’ Max Length=250
Name or FarmViewName	string	name of the Farm View. Max Length=50

Read-Only Fields

Field	Type	Description
ActiveDeviceCount	uint	Read-only count of active Devices in this Farm View. Default=0
DeviceCount	uint	Read-only count of Devices in this Farm View. Default=0
Guid or FarmViewId	Guid	Read-only GUID that uniquely identifies this Farm View.
MakActivateNeededCount	uint	Read-only count of active Devices that need MAK activation in this Farm View. Default=0

PvsGroup

Read-Only Fields

Field	Type	Description
Guid	Guid	GUID of the Active Directory group. 00000000-0000-0000-0000-000000000000 for Windows groups.
Name	string	Name of the Group.

PvsKeyRotationPendingServers

Read-Only Fields

Field	Type	Description
Active	string	1 if the Server is currently active, 2 if unknown, and 0 otherwise.
Name or ServerName	string	Computer name with no spaces.
SiteName	string	Name of the Site.

PvsLocalServer

Read-Only Field

Field	Type	Description
Name or LocalServer	string	NetBios name of local server.

PvsNewDiskVersion

Read-Only Fields

Field	Type	Description
Name	string	Name of the disk file without the extension.
Status	uint	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.

PvsScrambledDataReport

Read-Only Fields

Field	Type	Description
ScrambledData	string	The scrambled data field

Field	Type	Description
Value	string	The real value of the scrambled data field

PvsServer

Read/Write Fields

Field	Type	Description
AdMaxPasswordAge	uint	Number of days before a password expires. Min=1, Max=30, Default=7
AdMaxPasswordAgeEnabled	bool	Age the password, when set to true. Default=false
BootPauseSeconds	uint	Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10
BuffersPerThread	uint	Number of buffers per worker thread. Min=1, Max=128, Default=24
BusyDbConnectionRetryCount	uint	Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2
BusyDbConnectionRetryInterval	uint	Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25
Description	string	User description. Default=”” Max Length=250
EventLoggingEnabled	bool	Enable event logging, when set to true. Default=true

Field	Type	Description
FirstPort	uint	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
InitialQueryConnectionPoolSize	uint	Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50
InitialTransactionConnectionPoolSize	uint	Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50
IoBurstSize	uint	Number of bytes read/writes can send in a burst of packets. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=4096, Max=61440, Default=32768
Ip	System.Net.IPAddress[]	One or more streaming IPv4 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv4 addresses
Ipv6	string[]	One or more streaming IPv6 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv6 addresses
LastBugReportAttempt	DateTime	Time that this server last attempted to upload or generate a bug report bundle. Default=Empty

Field	Type	Description
LastBugReportResult	string	Status of the last bug report on this server. Default=’”’Max Length=4000
LastBugReportStatus	string	Status of the last bug report on this server. Default=’”’Max Length=250
LastBugReportSummary	string	Summary of the last bug report on this server. Default=’”’Max Length=250
LastCeipUploadAttempt	DateTime	Time that this server last attempted a CEIP upload. Default=Empty
LastPort	uint	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
LicenseTimeout	uint	Amount of seconds before a license times out. Min=15, Max=300, Default=30
LocalConcurrentIoLimit	uint	Maximum concurrent IO transactions it performs for vDisks that are local. A value of 0 disables the feature. Min=0, Max=128, Default=4
LogFileBackupCopiesMax	uint	Maximum number of log file backups. Min=1, Max=50, Default=4
LogFileSizeMax	uint	Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5

Field	Type	Description
LogLevel	uint	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
MaxBootDevicesAllowed	uint	Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500
MaxBootSeconds	uint	Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60
MaxQueryConnectionPoolSize	uint	Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000
MaxTransactionConnectionPoolSize	uint	Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000
MaxTransmissionUnits	uint	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
NonBlockingIoEnabled	bool	Use non-Blocking IO, when set to true. Default=true

Field	Type	Description
PowerRating	float	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
QuicCipherSuites	uint	QuicCipherSuites Default=''
QuicPort	uint	QuicPort Default=0
QuicServerCertThumbPrint	string	QuicServerCertThumbPrint Default=''Max Length=65535
QuicSettings	string	QuicSettings Default=''Max Length=65535
QuicStreamPayloadLength	uint	QuicStreamPayloadLength Default=0
RefreshInterval	uint	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
RemoteConcurrentIoLimit	uint	Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4
SecurityPolicy	uint	SecurityPolicy Default=0
ServerCacheTimeout	uint	Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8
ServerEpoch	UInt64	Timestamp when the registry data for that server has been re-encrypted with updated encryption scheme. Default=0

Field	Type	Description
Name or ServerName	string	Computer name with no spaces. ASCII computer name characters Max Length=21
ThreadsPerPort	uint	Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8
TpmCurveId	uint	TPM ECC curve ID TPM_ECC_NIST_P256(3), TPM_ECC_NIST_P384(4), TPM_ECC_NIST_P521(5) Default=0
TpmHashAlg	uint	TPM hash algorithm, TPM_ALG_SHA256(0xb), TPM_ALG_SHA384(0xc), TPM_ALG_SHA512(0xd) Default=0
TpmKeyType	uint	TPM key type, TPM_ALG_RSA(1), TPM_ALG_ECC(0x23) Default=0
UnusedDbConnectionTimeout	uint	Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300
VDiskCreatePacing	uint	VDisk create time pacing in milliseconds. Min=0, Max=5, Default=0

Read-Only Fields

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

Field	Type	Description
CloudSetupActive	bool	True if farm is integrated with Citrix Cloud and this PVS server has been registered with Citrix Cloud. Default=false
IdentityPublicKey	string	PVS Server's Identity Public key. This field is used for authentication in Peer-2-Peer communication between Servers. Default=''Max Length=65535
ManagementIp	System.Net.IPAddress	IP address used for management communications between Servers. Default=0.0.0.0
Role	uint	Read-only Role of the user for this item. 100 is Farm Administrator, and 200 is Site Administrator. Default=999
ServerFqdn	string	Read-only fully qualified domain name. Default=''Max Length=1024
Guid or ServerId	Guid	Read-only GUID that uniquely identifies this Server.
ServerRegistrationKeyStatus	uint	Status of the server registration key, 0 if valid, 1 if invalid, 2 if not present and 3 if failed to validate the registration key. Min=0, Max=4, Default=2
SiteId	Guid	GUID of the Site this Server is to be a member of. It is not used with SiteName.
SiteName	string	Name of the Site this Server is to be a member of. It is not used with SiteId.

Field	Type	Description
TrustCertificate	string	A PEM-encoded certificate that will be used to establish trust for the server's certificate for PvsRpc communication. Default=''Max Length=65535
XdControllerAddress	string	The Citrix Virtual Apps and Desktops controller used to register this PVS server with a CVAD site. Default=''Max Length=1024
XdSetupActive	bool	True if farm is integrated with Citrix Virtual Apps and Desktops and this PVS server has been registered with the CVAD site. Default=false

PvsServerBiosBootstrap

Read/Write Fields

Field	Type	Description
BootFromHdOnFail	bool	For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false
Bootserver1_Ip	System.Net.IPAddress	1st boot server IP. Only used when Lookup is false.
Bootserver1_Port	uint	1st boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Bootserver2_Ip	System.Net.IPAddress	2nd boot server IP. Only used when Lookup is false. Default=0.0.0.0

Field	Type	Description
Bootserver2_Port	uint	2nd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Bootserver3_Ip	System.Net.IPAddress	3rd boot server IP. Only used when Lookup is false. Default=0.0.0.0
Bootserver3_Port	uint	3rd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Bootserver4_Ip	System.Net.IPAddress	4th boot server IP. Only used when Lookup is false. Default=0.0.0.0
Bootserver4_Port	uint	4th boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
DhcpEnabled	bool	Use DHCP to retrieve target device IP when set to true, otherwise use the static domain, dnsIpAddresstrue and dnsIpAddress2 settings. Default=true
DnsIpAddress1	System.Net.IPAddress	Primary DNS server IP. Only used when DhcpEnabled is false.
DnsIpAddress2	System.Net.IPAddress	Secondary DNS server IP. Only used when DhcpEnabled is false.
Domain	string	Domain of the primary and secondary DNS servers. Only used when DhcpEnabled is false.

Field	Type	Description
Enabled	bool	Automatically update the BIOS on the target device with these setting when set to true, otherwise do not use these settings. Default=false
GeneralTimeout	uint	Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000
InterruptSafeMode	bool	Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false
Lookup	bool	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
PaeMode	bool	PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false
PollingTimeout	uint	Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000
RecoveryTime	uint	When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50

Field	Type	Description
Name or ServerName	string	Host to use for DNS lookup. Only used when Lookup is true. Default=IMAGESERVER1
VerboseMode	bool	Display verbose diagnostic information when set to true. Default=false

Read-Only Field

Field	Type	Description
Guid or ServerId	Guid	GUID of the Server used for Get and Set.

PvsServerBootstrap

Read/Write Fields

Field	Type	Description
BootFromHdOnFail	bool	For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false
Bootserver1_Gateway	System.Net.IPAddress	1st boot server gateway. Default=0.0.0.0
Bootserver1_Ip	System.Net.IPAddress	1st boot server IP.
Bootserver1_Netmask	System.Net.IPAddress	1st boot server netmask. Default=0.0.0.0
Bootserver1_Port	uint	1st boot server port. Min=1025, Max=65536, Default=6910
Bootserver2_Gateway	System.Net.IPAddress	2nd boot server gateway. Default=0.0.0.0
Bootserver2_Ip	System.Net.IPAddress	2nd boot server IP. Default=0.0.0.0
Bootserver2_Netmask	System.Net.IPAddress	2nd boot server netmask. Default=0.0.0.0

Field	Type	Description
Bootserver2_Port	uint	2nd boot server port. Min=1025, Max=65536, Default=6910
Bootserver3_Gateway	System.Net.IPAddress	3rd boot server gateway. Default=0.0.0.0
Bootserver3_Ip	System.Net.IPAddress	3rd boot server IP. Default=0.0.0.0
Bootserver3_Netmask	System.Net.IPAddress	3rd boot server netmask. Default=0.0.0.0
Bootserver3_Port	uint	3rd boot server port. Min=1025, Max=65536, Default=6910
Bootserver4_Gateway	System.Net.IPAddress	4th boot server gateway. Default=0.0.0.0
Bootserver4_Ip	System.Net.IPAddress	4th boot server IP. Default=0.0.0.0
Bootserver4_Netmask	System.Net.IPAddress	4th boot server netmask. Default=0.0.0.0
Bootserver4_Port	uint	4th boot server port. Min=1025, Max=65536, Default=6910
GeneralTimeout	uint	Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000
InterruptSafeMode	bool	Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false
PaeMode	bool	PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false
PollingTimeout	uint	Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000
RecoveryTime	uint	When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50

Field	Type	Description
VerboseMode	bool	Display verbose diagnostic information when set to true. Default=false

Read-Only Fields

Field	Type	Description
Name	string	Name of the bootstrap file used to Get and Set.
Guid or ServerId	Guid	GUID of the Server used for Get and Set.

PvsServerBootstrapName

Read-Only Field

Field	Type	Description
Name	string	Bootstrap file name.

PvsServerInfo

Read-Only Fields

Field	Type	Description
Active	uint	1 if the Server is currently active, 2 if unknown, and 0 otherwise. Min=0, Max=2, Default=0
AdMaxPasswordAge	uint	Number of days before a password expires. Min=1, Max=30, Default=7
AdMaxPasswordAgeEnabled	bool	Age the password, when set to true. Default=false

Field	Type	Description
BootPauseSeconds	uint	Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10
BuffersPerThread	uint	Number of buffers per worker thread. Min=1, Max=128, Default=24
BusyDbConnectionRetryCount	uint	Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2
BusyDbConnectionRetryInterval	uint	Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25
CloudSetupActive	bool	True if farm is integrated with Citrix Cloud and this PVS server has been registered with Citrix Cloud. Default=false
ContactIp	System.Net.IPAddress	Read-only contact IPv4 Address for the Server.
ContactIpv6	string	Read-only contact IPv6 Address for the Server.
ContactPort	string	Read-only contact port for the Server.
Description	string	User description. Default=’”’ Max Length=250
DeviceCount	uint	Read-only count of Devices. Default=0
EventLoggingEnabled	bool	Enable event logging, when set to true. Default=true
FirstPort	uint	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

Field	Type	Description
IdentityPublicKey	string	PVS Server's Identity Public key. This field is used for authentication in Peer-2-Peer communication between Servers. Default=""Max Length=65535
InitialQueryConnectionPoolSize	uint	Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50
InitialTransactionConnectionPoolSize	uint	Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50
IoBurstSize	uint	Number of bytes read/writes can send in a burst of packets. Required that IoBurstSize/(MaxTransmissionUnits-76)<=32. Min=4096, Max=61440, Default=32768
Ip	System.Net.IPAddress[]	One or more streaming IPv4 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv4 addresses
Ipv6	string[]	One or more streaming IPv6 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv6 addresses
LastBugReportAttempt	DateTime	Time that this server last attempted to upload or generate a bug report bundle. Default=Empty

Field	Type	Description
LastBugReportResult	string	Status of the last bug report on this server. Default=’”’Max Length=4000
LastBugReportStatus	string	Status of the last bug report on this server. Default=’”’Max Length=250
LastBugReportSummary	string	Summary of the last bug report on this server. Default=’”’Max Length=250
LastCeipUploadAttempt	DateTime	Time that this server last attempted a CEIP upload. Default=Empty
LastPort	uint	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
LicenseTimeout	uint	Amount of seconds before a license times out. Min=15, Max=300, Default=30
LocalConcurrentIoLimit	uint	Maximum concurrent IO transactions it performs for vDisks that are local. A value of 0 disables the feature. Min=0, Max=128, Default=4
LogFileBackupCopiesMax	uint	Maximum number of log file backups. Min=1, Max=50, Default=4
LogFileSizeMax	uint	Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5

Field	Type	Description
LogLevel	uint	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
ManagementIp	System.Net.IPAddress	IP address used for management communications between Servers. Default=0.0.0.0
MaxBootDevicesAllowed	uint	Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500
MaxBootSeconds	uint	Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60
MaxQueryConnectionPoolSize	uint	Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000
MaxTransactionConnectionPoolSize	uint	Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000
MaxTransmissionUnits	uint	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

Field	Type	Description
NonBlockingIoEnabled	bool	Use non-Blocking IO, when set to true. Default=true
PowerRating	float	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
QuicCipherSuites	uint	QuicCipherSuites Default=''
QuicPort	uint	QuicPort Default=0
QuicServerCertThumbPrint	string	QuicServerCertThumbPrint Default=''Max Length=65535
QuicSettings	string	QuicSettings Default=''Max Length=65535
QuicStreamPayloadLength	uint	QuicStreamPayloadLength Default=0
RefreshInterval	uint	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
RemoteConcurrentIoLimit	uint	Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4
Role	uint	Read-only Role of the user for this item. 100 is Farm Administrator, and 200 is Site Administrator. Default=999
SecurityPolicy	uint	SecurityPolicy Default=0
ServerCacheTimeout	uint	Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8

Field	Type	Description
ServerEpoch	UInt64	Timestamp when the registry data for that server has been re-encrypted with updated encryption scheme. Default=0
ServerFqdn	string	Read-only fully qualified domain name. Default=’”’Max Length=1024
Guid or ServerId	Guid	Read-only GUID that uniquely identifies this Server.
Name or ServerName	string	Computer name with no spaces. ASCII computer name characters Max Length=21
ServerRegistrationKeyStatus	uint	Status of the server registration key, 0 if valid, 1 if invalid, 2 if not present and 3 if failed to validate the registration key. Min=0, Max=4, Default=2
SiteId	Guid	GUID of the Site this Server is to be a member of. It is not used with SiteName.
SiteName	string	Name of the Site this Server is to be a member of. It is not used with SiteId.
ThreadsPerPort	uint	Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8
TpmCurveId	uint	TPM ECC curve ID TPM_ECC_NIST_P256(3), TPM_ECC_NIST_P384(4), TPM_ECC_NIST_P521(5) Default=0
TpmHashAlg	uint	TPM hash algorithm, TPM_ALG_SHA256(0xb), TPM_ALG_SHA384(0xc), TPM_ALG_SHA512(0xd) Default=0

Field	Type	Description
TpmKeyType	uint	TPM key type, TPM_ALG_RSA(1), TPM_ALG_ECC(0x23) Default=0
TrustCertificate	string	A PEM-encoded certificate that will be used to establish trust for the server's certificate for PvsRpc communication. Default=""Max Length=65535
UnusedDbConnectionTimeout	uint	Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300
VDiskCreatePacing	uint	VDisk create time pacing in milliseconds. Min=0, Max=5, Default=0
XdControllerAddress	string	The Citrix Virtual Apps and Desktops controller used to register this PVS server with a CVAD site. Default=""Max Length=1024
XdSetupActive	bool	True if farm is integrated with Citrix Virtual Apps and Desktops and this PVS server has been registered with the CVAD site. Default=false

PvsServerStatus

Read-Only Fields

Field	Type	Description
DeviceCount	uint	Read-only count of Devices. Default=0
Ip	System.Net.IPAddress	Read-only contact IPv4 Address for the Server.

Field	Type	Description
Ipv6	string	Read-only contact IPv6 Address for the Server. Default=0
Port	uint	Read-only contact port for the Server.
Guid or ServerId	Guid	Read-only GUID of the Server. Can be used with Get Server.
Name or ServerName	string	Read-only Name of the Server. Can be used with Get Server.
Status	uint	Status of the server, 0 if down, 1 if up and 2 if unknown. Default=0

PvsServerStore

Read/Write Fields

Field	Type	Description
CachePath	string[]	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
Path	string	Directory path that the Server uses to access the Store. Default=”“Max Length=255

Read-Only Fields

Field	Type	Description
ServerId	Guid	GUID of the server that uses the Store. ServerName can be used instead.
ServerName	string	Name of the server that uses the Store. ServerId can be used instead.

Field	Type	Description
StoreId	Guid	GUID of the Store. StoreName can be used instead.
StoreName	string	Name of the Store. StoreId can be used instead.

PvsSite

Read/Write Fields

Field	Type	Description
DefaultCollectionId	Guid	GUID of the Collection to place new Devices into automatically. Not used with defaultCollectionName. Default=00000000-0000-0000-0000-000000000000
DefaultCollectionName	string	Name of the Collection to place new Devices into automatically. Not used with DefaultCollectionId. Default=’”’
Description	string	User description. Default=’”’ Max Length=250
DiskUpdateServerId	Guid	GUID of the Disk Update Server for the Site. Not used with DiskUpdateServerName. Default=00000000-0000-0000-0000-000000000000
DiskUpdateServerName	string	Name of the Disk Update Server for the Site. Not used with DiskUpdateServerId. Default=’”’
EnableDiskUpdate	bool	True when Disk Updated is enabled for the Site, false otherwise. Default=false

Field	Type	Description
InventoryFilePollingInterval	uint	The number of seconds between polls for Disk changes in the Stores. Min=1, Max=600, Default=60
MakPassword	string	User password used for MAK activation. Default=’”’Max Length=64
MakUser	string	User name used for MAK activation. Default=’”’Max Length=64
Name or SiteName	string	Name of the Site. Max Length=50

Read-Only Fields

Field	Type	Description
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0
ResourceLocationId	string	The Citrix Cloud resource location ID if the farm SetupType is cloud mode. Default=’”’Max Length=64
ResourceLocationName	string	The Citrix Cloud resource location name if the farm SetupType is cloud mode. Default=’”’Max Length=128
Role	uint	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 or SiteId	Guid	Read-only GUID that uniquely identifies this Site.

PvsSiteView

Read/Write Fields

Field	Type	Description
Description	string	User description. Default=’”’ Max Length=250
Name or SiteViewName	string	Name of the Site View. Max Length=50

Read-Only Fields

Field	Type	Description
ActiveDeviceCount	uint	Read-only count of active Devices in this Site View. Default=0
DeviceCount	uint	Read-only count of Devices in this Site View. Default=0
DeviceWithPVDCount	uint	Read-only count of Devices with Personal vDisk in this Site View. Default=0
MakActivateNeededCount	uint	Read-only count of active Devices that need MAK activation in this Site View. Default=0
Role	uint	Read-only Role of the user for this item. 100 is Farm Administrator, and 200 is Site Administrator. Default=999
SiteId	Guid	GUID of the Site this View is to be a member of. It is not used with SiteName.
SiteName	string	Name of the Site this View is to be a member of. It is not used with SiteId.
Guid or SiteViewId	Guid	Read-only GUID that uniquely identifies this Site View.

PvsStore

Read/Write Fields

Field	Type	Description
CachePath	string[]	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
Description	string	User description. Default=”” Max Length=250
Path	string	Default directory path that the Servers use to access this Store. Max Length=255
SiteId	Guid	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	string	Name of the Site where Administrators of that Site can change this Store. Not used for Farm Stores. SiteId can be used instead. Default=””
Name or StoreName	string	Name of the Store. Max Length=50

Read-Only Fields

Field	Type	Description
PathType	string	Read-only field indicating if the vdisks are on a server’s local hard disk or on a remote share.

Field	Type	Description
Role	uint	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 or StoreId	Guid	Read-only GUID that uniquely identifies this Store.

PvsStoreSharedOrServerPath

Read-Only Fields

Field	Type	Description
Path	string	Directory path that the Servers use to access this Store.
StoreId	Guid	GUID of the Store.
StoreName	string	Name of the Store.

PvsTask

Read-Only Fields

Field	Type	Description
Command	string	Command being processed. Default=’”’Max Length=50

Field	Type	Description
CommandType	string	Type of the command. Values are: Add, Delete, Get, Info, Run, RunWithReturn, Set and SetList. Default=’”’ Max Length=13
ExpirationTime	DateTime	Time the task record may be removed from the database if the task does not complete. Has the date and time without milliseconds.
Handle	uint	Handle to a running function.
Ip	System.Net.IPAddress	IPv4 Address of the remote host.
Ipv6	string	IPv6 Address of the remote host. Default=0
MapiException	string	Exception result in XML format. Default=’”’
Port	uint	Port number of the remote service.
Results	string	Result in XML format. Default=’”’
ServerFqdn	string	Qualified name of the server. Default=’”’Max Length=1024
SiteId	Guid	GUID of the Site that this Task is being processed in. Default=00000000-0000-0000-0000-000000000000
SiteName	string	Name of the Site that that this Task is being processed in.
StartTime	DateTime	Time the task was started. Has the date and time without milliseconds.

Field	Type	Description
State	uint	State of the Task. Values are: 0 (Processing), 1 (Cancelled), and 2 (Complete). Min=0, Max=2
TaskId	uint	Unique ID of the task.

PvsUndefinedDisk

Read-Only Fields

Field	Type	Description
VHDX	bool	If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false
Name	string	Name of the disk file without the extension.
Status	uint	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

Field	Type	Description
Date	uint[]	Days of the month. Numbers from 1-31 are the only valid values. This is used with Monthly Date recurrence. Default=’”’Max Length=83
DayMask	uint	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
Description	string	User description. Default=’”’ Max Length=250
Domain	string	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
Enabled	bool	True when it will be processed, false otherwise. Default=true
EsdType	string	Esd to use. Valid values are SCCM or WSUS. If no value, a custom script is run on the client. Default=’”’Max Length=50
Hour	uint	The hour of the day to perform the task. Min=0, Max=23, Default=0
Minute	uint	The minute of the hour to perform the task. Min=0, Max=59, Default=0

Field	Type	Description
MonthlyOffset	uint	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
OrganizationUnit	string	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	uint	Access to place the version in after the update has occurred. 0 = Production, 1 = Test, 2 = Maintenance. Min=0, Max=2, Default=0
PostUpdateScript	string	Script file to run after the update finishes. Default=''Max Length=255

Field	Type	Description
PostVmScript	string	Script file to run after the VM is unloaded. Default=’”’Max Length=255
PreUpdateScript	string	Script file to run before the update starts. Default=’”’Max Length=255
PreVmScript	string	Script file to run before the VM is loaded. Default=’”’Max Length=255
Recurrence	uint	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
Name or UpdateTaskName	string	Name of the Update Task. It is unique within the Site. Max Length=50

Read-Only Fields

Field	Type	Description
SiteId	Guid	GUID of the Site that this Update Task is a member of. It is not used with SiteName.
SiteName	string	Name of the Site that this Update Task is a member of. It is not used with SiteId.
Guid or UpdateTaskId	Guid	Read-only GUID that uniquely identifies this Update Task.

PvsVersion

Read-Only Fields

Field	Type	Description
DbEdition	string	Edition of the database. If 'Express Edition', monitor dbSize.
DbSize	uint	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
DbVersion	uint	Version of the database schema as a number. Default=0
MapiVersion	string	Version of the system in major.minor.point.build format.
MapiVersionNumber	uint	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
SdkVersion	string	Version of the SDK in major.minor.build format.
Type	uint	Type of system. Values are 0 (Normal), 1 (OROM), and 2 (Secure). Default=0

PvsVirtualHostingPool

Read/Write Fields

Field	Type	Description
CredentialsUid	string	UUID of any record associated with the credentials Default="" Max Length=65535

Field	Type	Description
Datacenter	string	Datacenter of the Virtual Hosting Pool. Default=’”’Max Length=250
DdcType	uint	Type of the DDC. 0 = Unset, 1 = Citrix Cloud, 2 = customer-managed. Min=0, Max=2, Default=0
Description	string	User description. Default=’”’ Max Length=250
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0
Password	string	Password to use when logging into the Server.
PlatformVersion	string	Hypervisor Host Version Default=’”’Max Length=250
Port	uint	Port of the Host Server. Min=80, Max=65534, Default=80
PrepopulateEnabled	bool	Enable prepopulate when set to true Default=false
Server	string	Name or IP of the Host Server. Max Length=255
ShutdownTimeout	uint	Timeout for shutdown. Min=2, Max=30, Default=10
Type	uint	Type of the Virtual Hosting Pool. 0 = Citrix XenServer, 1 = Microsoft SCVMM/Hyper-V, 2 = VMWare vSphere/ESX, 3 = Nutanix, 4 = Azure, 5 = GCP. Min=0, Max=255, Default=0
UpdateLimit	uint	Number of updates at the same time. Min=2, Max=1000, Default=1000
UpdateTimeout	uint	Timeout for updates. Min=2, Max=240, Default=60
UserName	string	Name to use when logging into the Server.

Field	Type	Description
Name or VirtualHostingPoolName	string	Name of the Virtual Hosting Pool. It is unique within the Site. Max Length=50
XdHcCustomProperties	string	Custom Properties for HCL Connection Details object Default=’”’Max Length=65535
XdHcHypervisorConnectionName	string	Hypervisor Connection Name for HCL Connection Details object Default=’”’Max Length=250
XdHcHypervisorConnectionUid	string	Hypervisor Connection Uid for HCL Connection Details object Default=’”’Max Length=250
XdHcRevision	string	Revision for HCL Connection Details object Default=’”’Max Length=250
XdHcSslThumbprints	string	Ssl Thumbprints for HCL Connection Details object Default=’”’Max Length=65535
XdHostingUnitUuid	Guid	UUID of XenDesktop Hosting Unit Default=00000000-0000-0000-0000-000000000000
XsPvsSiteUuid	Guid	UUID of XenServer PVS_site Default=00000000-0000-0000-0000-000000000000

Read-Only Fields

Field	Type	Description
SiteId	Guid	GUID of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteName.
SiteName	string	Name of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteId.

Field	Type	Description
Guid or VirtualHostingPoolId	Guid	Read-only GUID that uniquely identifies this Virtual Hosting Pool.

PvsXDSite

Read/Write Field

Field	Type	Description
ConfigServices	string[]	XenDesktop Server addresses. Max Length=2000

Read-Only Field

Field	Type	Description
Guid or XdSiteId	Guid	GUID of the XenDesktop Site.

Cmdlets

January 13, 2026

Add-PvsDeviceToDomain

Add a Device, all Devices in a Collection or View to a Domain.

One of these is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Add to the Domain.
Name or DeviceName	string[]	Name of the Device to Add to the Domain.

Name	Type	Description
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Add to the Domain.
CollectionId	Guid[]	GUID of the Collection to Add all Devices to the Domain.
SiteViewId	Guid[]	GUID of the Site View to Add all Devices to the Domain.
FarmViewId	Guid[]	GUID of the Farm View to Add all Devices to the Domain.
FarmViewName	string[]	Name of the Farm View to Add all Devices to the Domain.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string[]	
SiteViewName	string[]	

Optional

Name	Type	Description
Domain	string[]	
OrganizationUnit	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Move.
Name or DeviceName	string[]	Name of the Device to Move.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Move.

One of these is required

Name	Type	Description
SiteViewId	Guid[]	
FarmViewId	Guid[]	
FarmViewName	string[]	

or this is required & resolution

Name	Type	Description
SiteViewName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Assign.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

One of these is required

Name	Type	Description
DeviceId	Guid[]	
DeviceName	string[]	
DeviceMac	PvsPhysicalAddress[]	
CollectionId	Guid[]	
SiteViewId	Guid[]	
FarmViewId	Guid[]	
FarmViewName	string[]	

or one of these is required & resolutions

Name	Type	Description
CollectionName	string[]	
SiteViewName	string[]	

Optional

Name	Type	Description
RemoveExisting	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Assign.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

One of these is required

Name	Type	Description
UpdateTaskId	Guid[]	
UpdateTaskName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator File to Add the new Disk Version(s) to.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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

Name	Type	Description
Persist	SwitchParameter	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.

Examples

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.

These are optional

Name	Type	Description
TaskId	uint	Id of the Task to Clear.
SiteId	Guid[]	Site Id of the Tasks to Clear.
SiteName	string[]	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 pipeline:
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.

Examples

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.

```
1  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 is required

Name	Type	Description
DeviceIdFrom	Guid	GUID of the Device to Copy from.
DeviceNameFrom	string	Name of the Device to Copy from.
DeviceMacFrom	PvsPhysicalAddress	Mac of the Device to Copy from.

One of these is required

Name	Type	Description
Guid or DeviceId	Guid	
Name or DeviceName	string	
DeviceMac	PvsPhysicalAddress	
CollectionId	Guid	
SiteViewId	Guid	
FarmViewId	Guid	
FarmViewName	string	

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	

Optional

Name	Type	Description
Properties	uint[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
DiskLocatorIdFrom	Guid	GUID of the Disk Locator to Copy from.

This is required

Name	Type	Description
Guid or DiskLocatorId	Guid	

Optional

Name	Type	Description
Properties	uint[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
ServerIdFrom	Guid	GUID of the Server to Copy from.
ServerNameFrom	string	Name of the Server to Copy from.

One of these is required

Name	Type	Description
Guid or ServerId	Guid	
Name or ServerName	string	

Optional

Name	Type	Description
Properties	uint[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Disable the DiskLocator for.
Name or DeviceName	string[]	Name of the Device to Disable the DiskLocator for.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Disable the DiskLocator for.

This is required

Name	Type	Description
DiskLocatorId	Guid[]	

or this is required & resolution

Name	Type	Description
DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Enable the DiskLocator for.
Name or DeviceName	string[]	Name of the Device to Enable the DiskLocator for.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Enable the DiskLocator for.

This is required

Name	Type	Description
DiskLocatorId	Guid[]	

or this is required & resolution

Name	Type	Description
DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

EXAMPLE 1: Enable a PvsDevice PvsDiskLocator This example enables the PvsDiskLocator named theDiskLocator for the PvsDevice named theDevice.

```
1  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 is required

Name	Type	Description
FileName	string[]	Name of the file to archive the Audit Trail to. This must be a full file path name.

Optional

Name	Type	Description
EndDate	DateTime	
PurgeData	SwitchParameter	

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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator that identifies the disk to export.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

Optional

Name	Type	Description
Version	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
DiskLocatorId

Examples

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 is required

Name	Type	Description
FileName	string	Name of the file to export the Oem Licenses to. This must be a full file path name.

One of these is required

Name	Type	Description
Guid or DeviceId	Guid	
Name or DeviceName	string	
DeviceMac	PvsPhysicalAddress	
CollectionId	Guid	
SiteViewId	Guid	
FarmViewId	Guid	

Name	Type	Description
FarmViewName	string	

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

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 pipeline:

DeviceId, CollectionId, SiteViewId or FarmViewId

Examples

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 is required

Name	Type	Description
Domain	string	Domain the account is a member of.

This is required

Name	Type	Description
Name	string	

PvsADAccount: If successful, the PvsADAccount object is returned.

Examples

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 is required

Name	Type	Description
AuditActionId	Guid[]	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 pipeline:

AuditActionId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Examples

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 is required

Name	Type	Description
AuditActionId	Guid[]	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 pipeline:
AuditActionId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Examples

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 is required

Name	Type	Description
Guid or AuditActionId	Guid[]	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 pipeline:
AuditActionId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
Guid or AuditActionId		GUID of the action.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsAuditActionSibling Get all PvsAuditActionSibling for the AuditActionId e8baa554-7c2d-49e5-9f6b-e0bc46179fc7.

```
1  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.

These are optional

Name	Type	Description
Guid or AuditActionId	Guid[]	GUID of the Audit Action to Get.
ParentId	Guid[]	Parent AuditActionId of the records to retrieve. If no parameters are included, only records with no parent are returned.
RootId	Guid[]	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.
SiteId	Guid[]	GUID of the Site to get the Audit Trail for.
SiteName	string[]	Name of the Site to get the Audit Trail for.

Name	Type	Description
CollectionId	Guid[]	GUID of the Collection to get the Audit Trail for.
SiteViewId	Guid[]	GUID of the Site View to get the Audit Trail for.
FarmViewId	Guid[]	GUID of the Farm View to get the Audit Trail for.
FarmViewName	string[]	Name of the Farm View to get the Audit Trail for.
ServerId	Guid[]	GUID of the Server to get the Audit Trail for.
ServerName	string[]	Name of the Server to get the Audit Trail for.
DeviceId	Guid[]	GUID of the Device to get the Audit Trail for.
DeviceName	string[]	Name of the Device to get the Audit Trail for.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to get the Audit Trail for.
StoreId	Guid[]	GUID of the Store to get the Audit Trail for.
StoreName	string[]	Name of the Store to get the Audit Trail for.
DiskLocatorId	Guid[]	GUID of the DiskLocator to get the Audit Trail for.

These are optional & resolutions

Name	Type	Description
CollectionName	string[]	
SiteViewName	string[]	
DiskLocatorName	string[]	

Optional

Name	Type	Description
UserName	string[]	
Domain	string[]	
BeginDate	DateTime	
EndDate	DateTime	
Type	uint[]	
Action	uint[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to get the Audit Trail for.
SiteName	string[]	Name of the Site to get the Audit Trail for.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	GUID of the Store to get the Audit Trail for.
StoreName	string[]	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 pipeline:

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.

Name	Type	Description
Guid or AuditActionId		GUID of the action.

Name	Type	Description
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)

Name	Type	Description
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

Name	Type	Description
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

Name	Type	Description
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.

Examples

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.

```
1 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.

```
1 Get-PvsAuditTrail -DeviceMac "02-50-F2-00-00-01"
```

EXAMPLE 8: Get PvsAuditTrail for Server Get all PvsAuditTrail for the Server named theServer.

```
1 Get-PvsAuditTrail -ServerName theServer
```

EXAMPLE 9: Get PvsAuditTrail for Store Get all PvsAuditTrail for the Store named theStore.

```
1 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.

```
1 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.

```
1 Get-PvsAuditTrail -UserName theUser -DomainName theDomain
```

EXAMPLE 12: Get PvsAuditTrail for January Get all PvsAuditTrail for January.

```
1 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.

These are optional

Name	Type	Description
Guid or AuthGroupId	Guid[]	GUID of the AuthGroup to Get.
Name or AuthGroupName	string[]	Name of the AuthGroup to Get.
SiteId	Guid[]	GUID of the Site to Get all AuthGroups with Authorization for.
SiteName	string[]	Name of the Site to Get all AuthGroups with Authorization for.
CollectionId	Guid[]	GUID of the Collection to Get all AuthGroups with Authorization for.

or this is optional & resolution

Name	Type	Description
CollectionName	string[]	

Optional

Name	Type	Description
Farm	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all AuthGroups with Authorization for.
SiteName	string[]	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 pipeline:

AuthGroupId, SiteId or CollectionId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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

Name	Type	Description
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
IsReadOnly		Indicates whether the administrative role is read-only(true) or read-write(false). Default=false

PvsAuthGroup[]: If successful, the PvsAuthGroup object(s) are returned.

Examples

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 is required

Name	Type	Description
AuthGroupId	Guid[]	GUID of the AuthGroup to Get all items that are authorized for it.
Name or AuthGroupName	string[]	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 pipeline:
AuthGroupId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsAuthGroupUsage Get all PvsAuthGroupUsage for the AuthGroup named theAuthGroup.

```
Get-PvsAuthGroupUsage -Name theAuthGroup
```

Get-PvsCeipData

Get the CEIP configuration

Optional

Name	Type	Description
Uuid	string[]	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 pipeline:
Uuid

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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

Name	Type	Description
OneTimeUpload		1 to perform a one time upload. Default=0

PvsCeipData[]: If successful, the PvsCeipData object(s) are returned.

Examples

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.

Name	Type	Description
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=65535
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.

Examples

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.

These are optional

Name	Type	Description
Guid or CollectionId	Guid[]	GUID of the Collection to Get.
SiteId	Guid[]	GUID of the Site to Get all Collections for.
SiteName	string[]	Name of the Site to Get all Collections for.

or this is optional & resolution

Name	Type	Description
Name or CollectionName	string[]	

This is optional

Name	Type	Description
ProvisioningType	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all Collections for.

Name	Type	Description
SiteName	string[]	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 pipeline:

CollectionId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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=””

Name	Type	Description
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

Name	Type	Description
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 ((lenautoAddPre-fix+lenautoAddSuffix)+AutoAddNumberLeng 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
ProvisioningType		The provisioning facility which created the Devices in this Collection. 0 is PVS, 1 is Studio. Only writable when adding a colection. Min=0, Max=1, Default=0

PvsCollection[]: If successful, the PvsCollection object(s) are returned.

Examples

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.

Examples

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 is required

Name	Type	Description
Name	string	Name of the Disk file that is being created.

One of these is required

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:

StoreId

UInt32 or PvsDiskLocator: If not finished, the percent complete is returned in an UInt32. If finished and successful, the PvsDiskLocator is returned.

Examples

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.

These are optional

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Get.
Name or DeviceName	string[]	Name of Device to Get.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Get.
EkPub	string[]	TPM EK public key of the Device to Get
CollectionId	Guid[]	GUID of the Collection to Get all Devices for.
ServerId	Guid[]	GUID of the Server to Get all Devices for.
ServerName	string[]	Name of the Server to Get all Devices for.
DiskLocatorId	Guid[]	GUID of the DiskLocator to Get all Devices for.
SiteViewId	Guid[]	GUID of the Site View to Get all Devices for.
SiteId	Guid[]	GUID of the Site.
SiteName	string[]	Name of the Site.
FarmViewId	Guid[]	GUID of the Farm View to Get all Devices for.
FarmViewName	string[]	Name of the Farm View to Get all Devices for.
BdmBoot	string[]	Include only the BDM Devices when set to 1. PXE devices if set to 0. If not included, all Devices are returned.

These are optional & resolutions

Name	Type	Description
CollectionName	string[]	
DiskLocatorName	string[]	
SiteViewName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site.
SiteName	string[]	Name of the Site.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

DeviceName, CollectionId, ServerId, DiskLocatorId, SiteViewId, SiteId or FarmViewId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Name	Type	Description
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. Default=’”’
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

Name	Type	Description
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=65535
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

Name	Type	Description
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=65535
LogLevel		Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
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

Name	Type	Description
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=65535
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

Name	Type	Description
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		Timestamp when BDM device was created Default=Empty
XsPvsProxyUuid		UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000
WriteCacheDisk		Write cache disk number Default=’”’Max Length=3
WriteCachePartition		Write cache disk partition number Default=’”’Max Length=3
UpdatePageFileSettings		Update pagefile settings Default=false
EncryptionEpoch		Timestamp of the encryption Key used to encrypt fields Default=0
ProvisioningType		The provisioning facility which created the device. 0 is PVS, 1 is Studio. Only writable when adding a device. Min=0, Max=1, Default=0
EkPub		Client TPM EK public key Default=’”’
SecurityPolicy		Client security policy Expected values are 0, 2 or 10, Default=null.

PvsDevice[]: If successful, the PvsDevice object(s) are returned.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device.
Name or DeviceName	string[]	Name of the Device.
DeviceMac	PvsPhysicalAddress[]	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 pipeline:
    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	Type	Description
Name or Bootstrap		Name of the bootstrap file. Max Length=259

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsDeviceBootstrap for Device Get all PvsDeviceBootstrap for the Device named theDevice.

```
1  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.

```
1  Get-PvsDeviceBootstrap -DeviceMac "02-50-F2-00-00-01"
```

Get-PvsDeviceCount

Get count of Devices in a Collection or View.

One of these is required

Name	Type	Description
CollectionId	Guid	GUID of the Collection to get the Device Count of.
SiteViewId	Guid	GUID of the Site View to get the Device Count of.
FarmViewId	Guid	GUID of the Farm View to get the Device Count of.
FarmViewName	string	Name of the Farm View to get the Device Count of.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

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 pipeline:
CollectionId, SiteViewId or FarmViewId

UInt32: If successful, the numeric value is returned

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid	Device GUID, to see if the DiskLocator for it is enabled.
Name or DeviceName	string	Device name, to see if the DiskLocator for it is enabled.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device, to see if the DiskLocator for it is enabled.

This is required

Name	Type	Description
DiskLocatorId	Guid	

or this is required & resolution

Name	Type	Description
DiskLocatorName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:

DeviceId or DiskLocatorId

String: If successful, the String value is returned.

Examples

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.

These are optional

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to get the temporary disk version information for.
Name or DeviceName	string[]	Name of the Device to get the temporary disk version information for.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to get the temporary disk version information for.
SiteId	Guid[]	GUID of the Site to get temporary disk version information for, and also resolution for DiskLocatorName.
SiteName	string[]	Name of the Site to get temporary disk version information for, and also resolution for DiskLocatorName.
DiskLocatorId	Guid[]	GUID of the Disk Locator to get temporary disk version information for.

or this is optional & resolution

Name	Type	Description
DiskLocatorName	string[]	

This is optional & resolution

Name	Type	Description
Version	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to get temporary disk version information for, and also resolution for DiskLocatorName.
SiteName	string[]	Name of the Site to get temporary disk version information for, and also resolution for DiskLocatorName.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

DeviceId, SiteId or DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsDeviceDiskTempVersion for Device Get the PvsDeviceDiskTempVersion for the Device named theDevice.

```
1  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.

These are optional

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Get.
Name or DeviceName	string[]	Name of Device to Get.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Get.
EkPub	string[]	TPM EK public key of the Device to Get
CollectionId	Guid[]	GUID of the Collection to Get all Devices for.

Name	Type	Description
ServerId	Guid[]	GUID of the Server to Get all Devices for.
ServerName	string[]	Name of the Server to Get all Devices for.
DiskLocatorId	Guid[]	GUID of the DiskLocator to Get all Devices for.
SiteViewId	Guid[]	GUID of the Site View to Get all Devices for.
SiteId	Guid[]	GUID of the Site.
SiteName	string[]	Name of the Site.
FarmViewId	Guid[]	GUID of the Farm View to Get all Devices for.
FarmViewName	string[]	Name of the Farm View to Get all Devices for.
BdmBoot	string[]	Include only the BDM Devices when set to 1. PXE devices if set to 0. If not included, all Devices are returned.

These are optional & resolutions

Name	Type	Description
CollectionName	string[]	
DiskLocatorName	string[]	
SiteViewName	string[]	

Optional

Name	Type	Description
OnlyActive	SwitchParameter	
MakLicenseActivated	uint	
Version	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site.
SiteName	string[]	Name of the Site.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

DeviceName, CollectionId, ServerId, DiskLocatorId, SiteViewId, SiteId or FarmViewId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Name	Type	Description
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. Default=’”’
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

Name	Type	Description
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=65535
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

Name	Type	Description
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=65535
LogLevel		Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
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

Name	Type	Description
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=65535
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

Name	Type	Description
BdmCreated		Timestamp when BDM device was created Default=Empty
XsPvsProxyUuid		UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000
WriteCacheDisk		Write cache disk number Default=’”’Max Length=3
WritePartition		Write cache disk partition number Default=’”’Max Length=3
UpdatePageFileSettings		Update pagefile settings Default=false
EncryptionEpoch		Timestamp of the encryption Key used to encrypt fields Default=0
ProvisioningType		The provisioning facility which created the device. 0 is PVS, 1 is Studio. Only writable when adding a device. Min=0, Max=1, Default=0
EkPub		Client TPM EK public key Default=’”’
SecurityPolicy		Client security policy Expected values are 0, 2 or 10, Default=null.
Ip		Read-only IPv4 Address of the Device. It is equal to 0.0.0.0 if the Device is not active or not using IPv4
Ipv6		Read-only IPv6 Address of the device. It is equal to “::”if the device is not active or not using IPv6 Default=0
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

Name	Type	Description
ServerIpConnection		Read-only IPv4 Address of the Server that the Device is using. It is equal to 0.0.0.0 if the Device is not active or not using IPv4.
ServerIpv6Connection		Read-only IPv6 Address of the Server that the Device is using. It is equal to “::”if the Device is not active or not using IPv6. Default=0
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

Name	Type	Description
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.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsDeviceInfo for Farm Get all PvsDeviceInfo for the Farm.

```
1 Get-PvsDeviceInfo
```

EXAMPLE 2: Get PvsDeviceInfo for FarmView Get all PvsDeviceInfo for the FarmView named the-FarmView.

```
1 Get-PvsDeviceInfo -FarmViewName theFarmView
```

EXAMPLE 3: Get PvsDeviceInfo for Site Get all PvsDeviceInfo for the Site named theSite.

```
1 Get-PvsDeviceInfo -SiteName theSite
```

EXAMPLE 4: Get PvsDeviceInfo for SiteView Get all PvsDeviceInfo for the SiteView named theSite-View in the Site named theSite.

```
1 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.

```
1 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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device.
Name or DeviceName	string[]	Name of the Device.
DeviceMac	PvsPhysicalAddress[]	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 pipeline:
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	Type	Description
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.

Examples

EXAMPLE 1: Get PvsDevicePersonality for Device Get all PvsDevicePersonality for the Device named theDevice.

```
1  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.

```
1  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.

These are optional

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Get status for.
Name or DeviceName	string[]	Name of Device to Get status for.

Name	Type	Description
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Get status for.
EkPub	string[]	TPM EK public key of the Device to Get
ServerId	Guid[]	GUID of the Server to Get all Device Status for.
ServerName	string[]	Name of the Server to Get all Device Status for.
DiskLocatorId	Guid[]	GUID of the Disk Locator to Get all DeviceStatus for.
CollectionId	Guid[]	GUID of the Collection to Get all DeviceStatus for.

These are optional & resolutions

Name	Type	Description
DiskLocatorName	string[]	
CollectionName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

DeviceId, ServerId, DiskLocatorId or CollectionId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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 IPv4 Address of the Device.
Ipv6		Read-only IPv6 Address of the device Default=0
ServerPortConnection		Read-only Port of the Server that the Device is using. Default=0
ServerIpConnection		Read-only IPv4 Address of the Server that the Device is using.
ServerIpv6Connection		Read-only IPv6 Address of the Server that the Device is using. Default=0
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

Name	Type	Description
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
QuicCipherSuite		Read-only The QUIC cipher suite that the Device is using for streaming. It is none if legacy protocol is used. Default=””
StreamProtocol		Read-only The stream protocol that the Device is using: 0(Legacy protocol), 1(QUIC protocol). Min=0, Max=1, Default=0

PvsDeviceStatus[]: If successful, the PvsDeviceStatus object(s) are returned.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid	GUID of the Server to get a list of Directories or Drives.
Name or ServerName	string	Name of the Server to get a list of Directories or Drives.

Optional

Name	Type	Description
Path	string	

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 pipeline:
ServerId

string[]: If successful, the array of directory names is returned.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
    DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Name	Type	Description
PrinterManagementEnabled		Invalid printers will be deleted from the Device when set to true.
WriteCacheType		0 (Private), 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), 9 (Cache in Device RAM with Overflow on Hard Disk), 10 (Private async), 11(Server persistent async), 12 (Cache in Device RAM with Overflow on Hard Disk async) Min=0, Max=12, Default=0
LicenseMode		0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0
AccelerateOfficeActivation		Run script to activate office automatically.
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

Name	Type	Description
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
EnableSetLocalTime		Enable setting the local time bias on the disk version.

PvsDisk[]: If successful, the PvsDisk object(s) are returned.

Examples

EXAMPLE 1: Get PvsDisk for DiskLocator Get the PvsDisk for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
1 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.

These are optional

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Get.
DeviceId	Guid[]	GUID of the Device to Get all DiskLocators for.
DeviceName	string[]	Name of the Device to Get all DiskLocators for.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Get all DiskLocators for.
ServerId	Guid[]	GUID of the Server to Get all DiskLocators for.
ServerName	string[]	Name of the Server to Get all DiskLocators for.
UpdateTaskId	Guid[]	GUID of the Update Task to Get all DiskLocators for.
SiteId	Guid[]	GUID of the Site to Get all DiskLocators for.
SiteName	string[]	Name of the Site to Get all DiskLocators for.

These are optional & resolutions

Name	Type	Description
Name or DiskLocatorName	string[]	
StoreId	Guid[]	
StoreName	string[]	
UpdateTaskName	string[]	

These are optional

Name	Type	Description
Single	SwitchParameter	
All	SwitchParameter	

Optional

Name	Type	Description
OnlyActive	SwitchParameter	
UpdateDevice	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all DiskLocators for.
SiteName	string[]	Name of the Site to Get all DiskLocators for.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

DiskLocatorId, DeviceId, ServerId, UpdateTaskId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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=’”’

Name	Type	Description
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

Name	Type	Description
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

Name	Type	Description
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), 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), 9 (Cache in Device RAM with Overflow on Hard Disk), 10 (Private async), 11(Server persistent async), 12 (Cache in Device RAM with Overflow on Hard Disk async) Min=0, Max=12, Default=0
LicenseMode		0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0
AccelerateOfficeActivation		Run script to activate office automatically.
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

Name	Type	Description
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
EnableSetLocalTime		Enable setting the local time bias on the disk version.
DeviceCount		Read-only count of Devices. Default=0

Name	Type	Description
Locked		True if the Disk is currently locked, false otherwise. Default=false

PvsDiskInfo[]: If successful, the PvsDiskInfo object(s) are returned.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Get Disk Version Inventory of.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

Optional

Name	Type	Description
Version	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
(DiskLocatorId and Version) or DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsDiskInventory for DiskLocator Get all PvsDiskInventory for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
1  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.

These are optional

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Get.
DeviceId	Guid[]	GUID of the Device to Get all DiskLocators for.
DeviceName	string[]	Name of the Device to Get all DiskLocators for.

Name	Type	Description
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Get all DiskLocators for.
ServerId	Guid[]	GUID of the Server to Get all DiskLocators for.
ServerName	string[]	Name of the Server to Get all DiskLocators for.
UpdateTaskId	Guid[]	GUID of the Update Task to Get all DiskLocators for.
SiteId	Guid[]	GUID of the Site to Get all DiskLocators for.
SiteName	string[]	Name of the Site to Get all DiskLocators for.

These are optional & resolutions

Name	Type	Description
Name or DiskLocatorName	string[]	
StoreId	Guid[]	
StoreName	string[]	
UpdateTaskName	string[]	

These are optional

Name	Type	Description
Single	SwitchParameter	
All	SwitchParameter	

Optional

Name	Type	Description
OnlyActive	SwitchParameter	
UpdateDevice	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all DiskLocators for.
SiteName	string[]	Name of the Site to Get all DiskLocators for.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

DiskLocatorId, DeviceId, ServerId, UpdateTaskId, SiteId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Name	Type	Description
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

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsDiskLocator for Farm Get all PvsDiskLocator for the Farm.

```
1 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 is required

Name	Type	Description
SiteId	Guid	GUID of the Site to get the Disk Locator Count of.

Name	Type	Description
SiteName	string	Name of the Site to get the Disk Locator Count of.

These are optional

Name	Type	Description
Single	SwitchParameter	
All	SwitchParameter	

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 pipeline:
SiteId

UInt32: If successful, the numeric value is returned

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Get the Locks.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsDiskLocatorLock for DiskLocator Get all PvsDiskLocatorLock for the DiskLocator named theDiskLocator in the Site named theSite and Store named theStore.

```
1  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.

These are optional

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Disk Update Device to Get.
Name or DeviceName	string[]	Name of Disk Update Device to Get.

Name	Type	Description
DeviceMac	PvsPhysicalAddress[]	MAC of the Disk Update Device to Get.
ServerId	Guid[]	GUID of the Server to Get all Disk Update Devices for.
ServerName	string[]	Name of the Server to Get all Disk Update Devices for.
DiskLocatorId	Guid[]	GUID of the DiskLocator to Get the Disk Update Device for.
UpdateTaskId	Guid[]	GUID of the Update Task to Get all Disk Update Devices for.
SiteId	Guid[]	GUID of the Site.
SiteName	string[]	Name of the Site.

These are optional & resolutions

Name	Type	Description
DiskLocatorName	string[]	
UpdateTaskName	string[]	

Optional

Name	Type	Description
OnlyActive	SwitchParameter	
MakLicenseActivated	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site.
SiteName	string[]	Name of the Site.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

DeviceId, ServerId, DiskLocatorId, UpdateTaskId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Name	Type	Description
Description		User description. Default=’”’ Max Length=250
DeviceMac		Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device. Default=’”’
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), and 5 (Debug). Min=0, Max=6, Default=4
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

Name	Type	Description
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=65535
Ip		Read-only IPv4 Address of the Device. It is equal to 0.0.0.0 if the Device is not active or not using IPv4
Ipv6		Read-only IPv6 Address of the device. It is equal to "::"if the device is not active or not using IPv6 Default=0
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 IPv4 Address of the Server that the Device is using. It is equal to 0.0.0.0 if the Device is not active or not using IPv4.
ServerIpv6Connection		Read-only IPv6 Address of the Server that the Device is using. It is equal to "::"if the Device is not active or not using IPv6. Default=0

Name	Type	Description
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.

Name	Type	Description
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.

Examples

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.

These are optional

Name	Type	Description
UpdateTaskId	Guid[]	GUID of the Update Task to Get.
DeviceId	Guid[]	GUID of the Disk Update Device to Get Disk Update Status for.
DeviceName	string[]	Name of the Disk Update Device to Get Disk Update Status for.
DeviceMac	PvsPhysicalAddress[]	MAC of the Disk Update Device to Get Disk Update Status for.
Guid or DiskUpdateTaskId	Guid[]	GUID of the Disk Update Task and Device relationship to Get Disk Update Status for.
SiteId	Guid[]	GUID of the Site to Get all Update Tasks for. Also used with UpdateTaskName.
SiteName	string[]	Name of the Site to Get all Update Tasks for. Also used with UpdateTaskName.

or this is optional & resolution

Name	Type	Description
UpdateTaskName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all Update Tasks for. Also used with UpdateTaskName.
SiteName	string[]	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 pipeline:

UpdateTaskId, DeviceId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Name	Type	Description
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

Name	Type	Description
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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Get Disk Versions of.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

These are optional

Name	Type	Description
Version	uint	
Type	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

(DiskLocatorId and Version) or DiskLocatorId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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

Name	Type	Description
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=””
LocalTimeBias		The time zone set in the vDisk version as the offset from UTC in minutes. -1 indicates vDisk timezone is not known or applicable. Default is -1 Default=-1 Max Length=5
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

Name	Type	Description
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.

Examples

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 is required

Name	Type	Description
SiteName	string	Name of the Site.
ServerName	string	Server name, to see if it is already used in the Farm.
FarmViewName	string	Farm View name, to see if it is already used in the Farm.
DeviceName	string	Device name, to see if it is already used in the Farm.
DeviceMac	PvsPhysicalAddress[]	Device MAC, to see if it is already used in the Farm.

Name	Type	Description
StoreName	string	Store name, to see if it is already used.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	
VirtualHostingPoolName	string	
UpdateTaskName	string	
DiskLocatorName	string	

One of these resolutions when needed

Name	Type	Description
SiteName	string	Name of the Site.
SiteId	Guid	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	Store name, to see if it is already used.

Boolean: If successful, $true or $false is returned.

Examples

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

Name	Type	Description
Guid or FarmId	Guid[]	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 pipeline:
FarmId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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

Name	Type	Description
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
LicenseSKU		LicenseSKU. 0 for on-premises, 1 for cloud. Min=0, Max=1, Default=0
LicenseWebServicesPort		The license server web services for licensing port. Min=1025, Max=65534, Default=8083
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

Name	Type	Description
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
IsReadOnly		Indicates whether the administrative role is read-only(true) or read-write(false).
SetupType		Farm integration type. 0 for no integration, 1 for Citrix Cloud, 2 for Citrix Virtual Apps and Desktops. Min=0, Max=2, Default=0
CloudSetupActive		True if farm is integrated with Citrix Cloud and all PVS servers have been registered with Citrix Cloud. Default=false
CustomerId		The Citrix Cloud customer ID if the SetupType is Citrix Cloud. Default=”” Max Length=128

Name	Type	Description
CustomerName		The Citrix Cloud customer name if the SetupType is Citrix Cloud. Default=’”’Max Length=256
FarmEpoch		Timestamp when the Data Encryption Key (DEK) was created and stored in the Credential Wallet. Default=0
EncryptionStatus		The current status for database encryption. 0 = Idle, 1 = Distributing Keys, 2= Re-encrypting database Min=0, Max=2, Default=0
ManualKeyCyclingRequired		Indicates if a manual encryption key rotation is required(true) or not(false). Default=false
XdSetupActive		True if farm is integrated with Citrix Virtual Apps and Desktops and all PVS servers have been registered with the CVAD site. Default=false
XdSiteId		The CVAD site ID if the SetupType is Citrix Virtual Apps and Desktops. Default=00000000-0000-0000-0000-000000000000
XdSiteName		The CVAD site name if the SetupType is Citrix Virtual Apps and Desktops. Default=’”’Max Length=256
KeyRotationInitiatingServerFqdn		The FQDN of the Server who initiated the database encryption key rotation. Default=’”’Max Length=1024

Name	Type	Description
DefaultDeviceLogLevel		The default logging level used by newly created target devices. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
KeyExchangeType		The encryption key exchange type. 0=CredentialWallet, 1=Peer-2-Peer. Min=0, Max=1, Default=1
IsUsageDataScrambled		Indicates whether the sensitive information needs to be scrambled (true) or not (false) when uploading usage telemetry data. Default = true. Default=true
LicenseServerCertThumbprint		Stores the license server's server certificate thumbprint when the server certificate is explicitly accepted by the administrator. NULL if no certificate is explicitly accepted. Default=''Max Length=65535
EntitledState		Indicates the entitled state of PVS Farm with LAS. 0 (not entitled), 1 (entitled via LAS), 2 (entitled via V6)
EntitlementExpirationDate		Expiration date of the entitlement

PvsFarm[]: If successful, the PvsFarm object(s) are returned.

Examples

EXAMPLE 1: Get PvsFarm Get the PvsFarm.

```
1  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.

These are optional

Name	Type	Description
Guid or FarmViewId	Guid[]	GUID of the Farm View to Get.
Name or FarmViewName	string[]	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 pipeline:
FarmViewId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Examples

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

Name	Type	Description
Domain	string[]	Domain of user (may be the name of the local computer).
User	string[]	Name of user.
AdGroupsEnabled	SwitchParameter	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	Type	Description
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.

Examples

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-PvsKeyRotationPendingServers

Gets all the offline servers in Farm that have ServerEpoch different than FarmEpoch when EncryptionStatus is Distributing keys. When EncryptionStatus is Idle, it gets list of all offline servers.

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
Name or ServerName		Computer name with no spaces.
SiteName		Name of the Site.
Active		1 if the Server is currently active, 2 if unknown, and 0 otherwise.

PvsKeyRotationPendingServers[]: If successful, the PvsKeyRotationPendingServers object(s) are returned.

Examples

EXAMPLE 1: Get Get-PvsKeyRotationPendingServers Gets all the offline servers in Farm that have serverEpoch different than farmEpoch when encryptionStatus is Distributing keys. When encryption-Status is Idle, it gets list of all offline servers.

```
Get-PvsKeyRotationPendingServers
```

Get-PvsLocalServer

Return one record with the local server's NetBios name

| Name | Type | Description |
| :—| :—| :————- |

PvsLocalServer[]: If successful, the PvsLocalServer object(s) are returned.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid	DiskLocator GUID, to see if it has a maintenance version.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:

DiskLocatorId

Boolean: If successful, $true or $false is returned.

Examples

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 is required

Name	Type	Description
CollectionId	Guid	GUID of the Collection to get the Minimum LastAutoAddDeviceNumber for.

or this is required & resolution

Name	Type	Description
CollectionName	string	

Optional

Name	Type	Description
AutoAddPrefix	string	
AutoAddSuffix	string	
AutoAddNumberLength	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

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 pipeline:
CollectionId

UInt32: If successful, the numeric value is returned

Examples

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.

```
1  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.

These are optional

Name	Type	Description
ServerId	Guid	GUID of the Server.
ServerName	string	Name of the Server.

These are optional

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:
ServerId or StoreId

PvsDiskLocator: If successful, the mapped PvsDiskLocator is returned.

Examples

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.

Examples

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 is required

Name	Type	Description
ServerId	Guid[]	GUID of the Server to look for new Disk versions.
ServerName	string[]	Name of the Server to look for new Disk versions.

One of these is required

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

Optional

Name	Type	Description
AutoAddEnabled	SwitchParameter	

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 pipeline:
ServerId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
Name		Name of the disk file without the extension.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsNewDiskVersion Get all PvsNewDiskVersion in the Store named theStore using the Server named theServer.

```
1  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.

```
1  Get-PvsNewDiskVersion -ServerName theServer -StoreName theStore -
       AutoAddEnabled
```

Get-PvsScrambledDataReport

Returns a (scrambled value, real value) key pair for all device usage events (e.g. create, edit, delete)

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
ScrambledData		The scrambled data field
Value		The real value of the scrambled data field

PvsScrambledDataReport[]: If successful, the PvsScrambledDataReport object(s) are returned.

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.

These are optional

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to Get.
Name or ServerName	string[]	Name of the Server to Get.
SiteId	Guid[]	GUID of the Site to Get all Servers.
SiteName	string[]	Name of the Site to Get all Servers.
DiskLocatorId	Guid[]	GUID of the Disk Locator to Get all Servers.

or this is optional & resolution

Name	Type	Description
DiskLocatorName	string[]	

These are optional & resolutions

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

Optional

Name	Type	Description
All	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all Servers.
SiteName	string[]	Name of the Site to Get all Servers.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

ServerId, SiteId, DiskLocatorId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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

Name	Type	Description
LicenseTimeout		Amount of seconds before a license times out. Min=15, Max=300, Default=30
VDiskCreatePacing		VDisk create time pacing in milliseconds. 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

Name	Type	Description
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=true
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 IPv4 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv4 addresses

Name	Type	Description
Ipv6		One or more streaming IPv6 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv6 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

Name	Type	Description
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

Name	Type	Description
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
CloudSetupActive		True if farm is integrated with Citrix Cloud and this PVS server has been registered with Citrix Cloud. Default=false
ServerEpoch		Timestamp when the registry data for that server has been re-encrypted with updated encryption scheme. Default=0
ServerRegistrationKeyStatus		Status of the server registration key, 0 if valid, 1 if invalid, 2 if not present and 3 if failed to validate the registration key. Min=0, Max=4, Default=2

Name	Type	Description
XdSetupActive		True if farm is integrated with Citrix Virtual Apps and Desktops and this PVS server has been registered with the CVAD site. Default=false
IdentityPublicKey		PVS Server's Identity Public key. This field is used for authentication in Peer-2-Peer communication between Servers. Default=''Max Length=65535
XdControllerAddress		The Citrix Virtual Apps and Desktops controller used to register this PVS server with a CVAD site. Default=''Max Length=1024
SecurityPolicy		SecurityPolicy Default=0
QuicSettings		QuicSettings Default=''Max Length=65535
QuicPort		QuicPort Default=0
QuicServerCertThumbPrint		QuicServerCertThumbPrint Default=''Max Length=65535
QuicCipherSuites		QuicCipherSuites Default=''
QuicStreamPayloadLength		QuicStreamPayloadLength Default=0
TpmKeyType		TPM key type, TPM_ALG_RSA(1), TPM_ALG_ECC(0x23) Default=0
TpmHashAlg		TPM hash algorithm, TPM_ALG_SHA256(0xb), TPM_ALG_SHA384(0xc), TPM_ALG_SHA512(0xd) Default=0

Name	Type	Description
TpmCurveId		TPM ECC curve ID TPM_ECC_NIST_P256(3), TPM_ECC_NIST_P384(4), TPM_ECC_NIST_P521(5) Default=0
TrustCertificate		A PEM-encoded certificate that will be used to establish trust for the server's certificate for PvsRpc communication. Default=''Max Length=65535

PvsServer[]: If successful, the PvsServer object(s) are returned.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to Get the dell_bios.bin BIOS bootstrap file from.
Name or ServerName	string[]	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 pipeline:
ServerId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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

Name	Type	Description
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

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsServerBiosBootstrap for Server Get all PvsServerBiosBootstrap for the Server named theServer.

```
1  Get-PvsServerBiosBootstrap -Name theServer
```

Get-PvsServerBootstrap

Get the bootstrap fields for the Server and named bootstrap file specified.

One of these is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to Get the named bootstrap file from.
ServerName	string[]	Name of the Server to Get the named bootstrap file from.

This is required

Name	Type	Description
Name	string[]	

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 pipeline:
(ServerId and Name) or ServerId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsServerBootstrap Get the PvsServerBootstrap for the Bootstrap named the-Bootstrap on the Server named theServer.

```
1 Get-PvsServerBootstrap -ServerName theServer -Name theBootstrap
```


Get-PvsServerBootstrapName

Get the bootstrap names for a Server.

One of these is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to Get bootstrap names for.
Name or ServerName	string[]	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 pipeline:
ServerId

| Name | Type | Description |
| :—| :—| :———- |

PvsServerBootstrapName[]: If successful, the PvsServerBootstrapName object(s) are returned.

Examples

EXAMPLE 1: Get PvsServerBootstrapName for Server Get all PvsServerBootstrapName for the Server named theServer.

```
1  Get-PvsServerBootstrapName -Name theServer
```

Get-PvsServerCount

Get count of Servers in a Site.

One of these is required

Name	Type	Description
Guid or SiteId	Guid	GUID of the Site to get the Server Count of.
Name or SiteName	string	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 pipeline:
SiteId

UInt32: If successful, the numeric value is returned

Examples

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.

These are optional

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to Get.
Name or ServerName	string[]	Name of the Server to Get.
SiteId	Guid[]	GUID of the Site to Get all Servers.
SiteName	string[]	Name of the Site to Get all Servers.
DiskLocatorId	Guid[]	GUID of the Disk Locator to Get all Servers.

or this is optional & resolution

Name	Type	Description
DiskLocatorName	string[]	

These are optional & resolutions

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

Optional

Name	Type	Description
All	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all Servers.
SiteName	string[]	Name of the Site to Get all Servers.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

ServerId, SiteId, DiskLocatorId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
Guid or ServerId		Read-only GUID that uniquely identifies this Server.

Name	Type	Description
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 milliseconds. 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

Name	Type	Description
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=true
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

Name	Type	Description
Ip		One or more streaming IPv4 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv4 addresses
Ipv6		One or more streaming IPv6 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv6 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

Name	Type	Description
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

Name	Type	Description
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
CloudSetupActive		True if farm is integrated with Citrix Cloud and this PVS server has been registered with Citrix Cloud. Default=false

Name	Type	Description
ServerEpoch		Timestamp when the registry data for that server has been re-encrypted with updated encryption scheme. Default=0
ServerRegistrationKeyStatus		Status of the server registration key, 0 if valid, 1 if invalid, 2 if not present and 3 if failed to validate the registration key. Min=0, Max=4, Default=2
XdSetupActive		True if farm is integrated with Citrix Virtual Apps and Desktops and this PVS server has been registered with the CVAD site. Default=false
IdentityPublicKey		PVS Server's Identity Public key. This field is used for authentication in Peer-2-Peer communication between Servers. Default=''Max Length=65535
XdControllerAddress		The Citrix Virtual Apps and Desktops controller used to register this PVS server with a CVAD site. Default=''Max Length=1024
SecurityPolicy		SecurityPolicy Default=0
QuicSettings		QuicSettings Default=''Max Length=65535
QuicPort		QuicPort Default=0
QuicServerCertThumbPrint		QuicServerCertThumbPrint Default=''Max Length=65535
QuicCipherSuites		QuicCipherSuites Default=''
QuicStreamPayloadLength		QuicStreamPayloadLength Default=0

Name	Type	Description
TpmKeyType		TPM key type, TPM_ALG_RSA(1), TPM_ALG_ECC(0x23) Default=0
TpmHashAlg		TPM hash algorithm, TPM_ALG_SHA256(0xb), TPM_ALG_SHA384(0xc), TPM_ALG_SHA512(0xd) Default=0
TpmCurveId		TPM ECC curve ID TPM_ECC_NIST_P256(3), TPM_ECC_NIST_P384(4), TPM_ECC_NIST_P521(5) Default=0
TrustCertificate		A PEM-encoded certificate that will be used to establish trust for the server's certificate for PvsRpc communication. Default=''Max Length=65535
ContactIp		Read-only contact IPv4 Address for the Server.
ContactIpv6		Read-only contact IPv6 Address 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.

Examples

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.

Examples

EXAMPLE 1: Get-PvsServerName

```
Get-PvsServerName
```


Get-PvsServerStatus

Get the Server Status fields for a Server.

One of these is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to Get status for.
Name or ServerName	string[]	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 pipeline:
ServerId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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 IPv4 Address 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
Ipv6		Read-only contact IPv6 Address for the Server. Default=0

PvsServerStatus[]: If successful, the PvsServerStatus object(s) are returned.

Examples

EXAMPLE 1: Get PvsServerStatus for Server Get the PvsServerStatus for the Server named the-Server.

```
Get-PvsServerStatus -Name theServer
```

Get-PvsServerStore

Get the directory and cache paths of a Server for one or all Stores.

One of these is required

Name	Type	Description
ServerId	Guid[]	GUID of a Server.
ServerName	string[]	Name of a Server.

These are optional

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
(ServerId and StoreId), ServerId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
StoreId		GUID of the Store. StoreName can be used instead.
StoreName		Name of the Store. StoreId can be used instead.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsServerStore for Server Get all PvsServerStore for the Server named theServer.

```
1 Get-PvsServerStore -ServerName theServer
```

EXAMPLE 2: Get PvsServerStore for Server and Store Get the PvsServerStore for the Server named theServer and Store named theStore.

```
1 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 is required

Name	Type	Description
ServerId	Guid	GUID of the Server.
ServerName	string	Name of the Server.

One of these is required

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:
ServerId or StoreId

UInt32: If successful, the numeric value is returned

Examples

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.

These are optional

Name	Type	Description
Guid or SiteId	Guid[]	GUID of the Site to Get.
Name or SiteName	string[]	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 pipeline:
SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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 with DiskUpdateServerId. Default=’”’

Name	Type	Description
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
ResourceLocationId		The Citrix Cloud resource location ID if the farm SetupType is cloud mode. Default=’”’Max Length=64
ResourceLocationName		The Citrix Cloud resource location name if the farm SetupType is cloud mode. Default=’”’Max Length=128
EncryptionEpoch		Timestamp of the encryption Key used to encrypt fields Default=0

PvsSite[]: If successful, the PvsSite object(s) are returned.

Examples

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.

These are optional

Name	Type	Description
Guid or SiteViewId	Guid[]	GUID of the Site View to Get.
SiteId	Guid[]	GUID of the Site to Get all Views for.
SiteName	string[]	Name of the Site to Get all Views for.

or this is optional & resolution

Name	Type	Description
Name or SiteViewName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all Views for.
SiteName	string[]	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 pipeline:
SiteViewId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
Guid or SiteViewId		Read-only GUID that uniquely identifies this Site View.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsSiteView for Farm Get all PvsSiteView for the Farm.

```
1 Get-PvsSiteView
```

EXAMPLE 2: Get PvsSiteView for Site Get all PvsSiteView for the Site named theSite.

```
1 Get-PvsSiteView -SiteName theSite
```


EXAMPLE 3: Get PvsSiteView Get the PvsSiteView for the SiteView named theSiteView in the Site named theSite.

```
1 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.

These are optional

Name	Type	Description
Guid or StoreId	Guid[]	GUID of the Store to Get.
Name or StoreName	string[]	Name of the Store to Get.
ServerId	Guid[]	GUID of the Server to Get all Stores for.
ServerName	string[]	Name of the Server to Get all Stores for.
SiteId	Guid[]	GUID of the Site to Get all Stores for.
SiteName	string[]	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 pipeline:
StoreId, ServerId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
Guid or StoreId		Read-only GUID that uniquely identifies this Store.
Name or StoreName		Name of the Store. Max Length=50

Name	Type	Description
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.

Examples

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 is required & resolutions

Name	Type	Description
Guid or StoreId	Guid	GUID of the Store.
Name or StoreName	string	Name of the Store.

One of these resolutions when needed

Name	Type	Description
ServerId	Guid	
ServerName	string	
SiteId	Guid	
SiteName	string	

UInt32: If successful, the numeric value is returned

Examples

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 is required

Name	Type	Description
SiteId	Guid[]	GUID of a Site used for authorization check.
SiteName	string[]	Name of a Site used for authorization check.

Optional

Name	Type	Description
ServerName	string[]	

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 pipeline:
SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsStoreSharedOrServerPath with Sharded UNC Paths Get PvsStoreSharedOrServerPath with shared UNC paths for the Farm.

```
1 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.

```
1 Get-PvsStoreSharedOrServerPath -SiteName theSite -StoreName theServer
```

Get-PvsTask

Get the current Task fields for select, or all active and completed un-cleared tasks.

These are optional

Name	Type	Description
TaskId	uint	ID of the Task to get.
SiteId	Guid[]	GUID of the Site to get Tasks for.
SiteName	string[]	Name of the Site to get Tasks for.

Optional

Name	Type	Description
State	uint	

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 pipeline:
TaskId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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		IPv4 Address of the remote host.
Ipv6		IPv6 Address of the remote host. Default=0
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=’”’

Name	Type	Description
Results		Result in XML format. Default=’”’

PvsTask[]: If successful, the PvsTask object(s) are returned.

Examples

EXAMPLE 1: Get PvsTask for Farm Get all PvsTask for the Farm.

```
1 Get-PvsTask
```

EXAMPLE 2: Get PvsTask for Site Get all PvsTask for the Site named theSite.

```
1 Get-PvsTask -SiteName theSite
```

EXAMPLE 3: Get PvsTask Currently Processing Get PvsTask that are currently in Processing state for the Farm.

```
1 Get-PvsTask -State 0
```

EXAMPLE 4: Get PvsTask Get PvsTask for TaskId 101.

```
1 Get-PvsTask -TaskId 101
```

Get-PvsTaskStatus

Get the status of a Task in percent complete.

This is required

Name	Type	Description
TaskId	uint	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 pipeline:

TaskId

UInt32: If successful, the numeric value is returned

Examples

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 is required

Name	Type	Description
ServerId	Guid[]	GUID of the Server to look for undefined Disks.
ServerName	string[]	Name of the Server to look for undefined Disks.

One of these is required

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

Optional

Name	Type	Description
AutoAddEnabled	SwitchParameter	

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 pipeline:

ServerId or StoreId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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)

Name	Type	Description
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.

Examples

EXAMPLE 1: Get PvsUndefinedDisk Get all PvsUndefinedDisk in the Store named theStore using the Server named theServer.

```
1 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.

```
1 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.

These are optional

Name	Type	Description
Guid or UpdateTaskId	Guid[]	GUID of the Update Task to Get.
SiteId	Guid[]	GUID of the Site to Get all Update Tasks for.
SiteName	string[]	Name of the Site to Get all Update Tasks for.

or this is optional & resolution

Name	Type	Description
Name or UpdateTaskName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all Update Tasks for.
SiteName	string[]	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 pipeline:

UpdateTaskId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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

Name	Type	Description
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

Name	Type	Description
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

Name	Type	Description
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.

Examples

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 is optional

Name	Type	Description
OneTimeUpload	string	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.

Name	Type	Description
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.

Examples

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.

These are optional

Name	Type	Description
Guid or VirtualHostingPoolId	Guid[]	GUID of the Virtual Hosting Pool to Get.
SiteId	Guid[]	GUID of the Site to Get all Virtual Hosting Pools for.
SiteName	string[]	Name of the Site to Get all Virtual Hosting Pools for.

or this is optional & resolution

Name	Type	Description
Name or VirtualHostingPoolName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site to Get all Virtual Hosting Pools for.
SiteName	string[]	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 pipeline:

VirtualHostingPoolId or SiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
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, 3 = Nutanix, 4 = Azure, 5 = GCP. Min=0, Max=255, Default=0
Description		User description. Default=”” Max Length=250
Server		Name or IP of the Host Server. Max Length=255

Name	Type	Description
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

Name	Type	Description
XdHcCustomProperties		Custom Properties for HCL Connection Details object Default=’”’Max Length=65535
XdHcSslThumbprints		Ssl Thumbprints for HCL Connection Details object Default=’”’Max Length=65535
CredentialsUid		UUID of any record associated with the credentials Default=’”’ Max Length=65535
DdcType		Type of the DDC. 0 = Unset, 1 = Citrix Cloud, 2 = customer-managed. Min=0, Max=2, Default=0
EncryptionEpoch		Timestamp of the encryption Key used to encrypt fields Default=0

PvsVirtualHostingPool[]: If successful, the PvsVirtualHostingPool object(s) are returned.

Examples

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 is optional

Name	Type	Description
Guid or XdSiteId	Guid[]	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 pipeline:
XdSiteId

If only selected fields are needed, pass them in the Fields parameter as a string array.

Name	Type	Description
Guid or XdSiteId		GUID of the XenDesktop Site.
ConfigServices		XenDesktop Server addresses. Max Length=2000

PvsXDSite[]: If successful, the PvsXDSite object(s) are returned.

Examples

EXAMPLE 1: Get PvsXDSite for Farm Get all PvsXDSite for the Farm.

```
1  Get-PvsXDSite
```

EXAMPLE 2: Get PvsXDSite Get the PvsXDSite for the XDSite with Guid 45687f34-c9ec-4852-9d55-337a1af41405.

```
1  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 is required

Name	Type	Description
Guid or AuthGroupId	Guid[]	GUID of the AuthGroup to assign Authorization for.
Name or AuthGroupName	string[]	Name of the AuthGroup to assign Authorization for.

These are optional

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	
CollectionId	Guid[]	

or this is optional & resolution

Name	Type	Description
CollectionName	string[]	

This is optional

Name	Type	Description
IsReadOnly	string[]	

Optional

Name	Type	Description
Role	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Name or FileName	string[]	Name of the file to import the Devices from. This must be a full file path name.

These are optional

Name	Type	Description
CollectionId	Guid[]	
SiteId	Guid[]	
SiteName	string[]	

or this is optional & resolution

Name	Type	Description
CollectionName	string[]	

Optional

Name	Type	Description
CopyTemplate	SwitchParameter	
DoNotCreateNewSites	SwitchParameter	
DoNotCreateNewCollections	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:

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.

Examples

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

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

These are optional

Name	Type	Description
ServerId	Guid	
ServerName	string	

Optional

Name	Type	Description
Description	string	
MenuText	string	
Enabled	SwitchParameter	
RebalanceEnabled	SwitchParameter	
RebalanceTriggerPercent	uint	
SubnetAffinity	uint	
Format	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Name or FileName	string[]	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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Activate.
Name or DeviceName	string[]	Name of the Device to Activate.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Activate.

This is required

Name	Type	Description
MakUsedToActivate	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
DeviceId	Guid[]	GUID of the Device to Mark Down.
DeviceName	string[]	Name of the Device to Mark Down.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Mark Down.
CollectionId	Guid[]	GUID of the Collection to Mark Down all Devices.
SiteViewId	Guid[]	GUID of the Site View to Mark Down all Devices.
SiteId	Guid[]	GUID of the Site. Can be used alone to Mark Down all Servers and Devices in the Site.
SiteName	string[]	Name of the Site. Can be used alone to Mark Down all Servers and Devices in the Site.
FarmViewId	Guid[]	GUID of the Farm View to Mark Down all Devices.
FarmViewName	string[]	Name of the Farm View to Mark Down all Devices.
DiskLocatorId	Guid[]	GUID of the DiskLocator to Mark Down all Devices.
ServerId	Guid[]	GUID of the Server to Mark Down.
ServerName	string[]	Name of the Server to Mark Down.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string[]	
SiteViewName	string[]	

Name	Type	Description
DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site. Can be used alone to Mark Down all Servers and Devices in the Site.
SiteName	string[]	Name of the Site. Can be used alone to Mark Down all Servers and Devices in the Site.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator File to Promote the Disk Version of.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

Optional

Name	Type	Description
ScheduledDate	DateTime	
TestVersion	uint	
Test	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid	GUID of the Server to Rebalance Devices on, ServerName.
Name or ServerName	string	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 pipeline:
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

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator File to Revert.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

Optional

Name	Type	Description
Version	uint	
Test	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator File to Merge.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

Optional

Name	Type	Description
Access	uint	
Base	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid	GUID of the Disk Locator to Mount the Disk.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string	

These are optional

Name	Type	Description
ServerId	Guid	
ServerName	string	

Optional

Name	Type	Description
MaxDiskLetterWaitSeconds	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Move.
Name or DeviceName	string[]	Name of the Device to Move.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Move.

This is required

Name	Type	Description
CollectionId	Guid[]	

or this is required & resolution

Name	Type	Description
CollectionName	string[]	

Optional

Name	Type	Description
CopyTemplate	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to Assign.
Name or ServerName	string[]	Name of the Server to Assign.

One of these is required

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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.

Name	Type	Description
Name or AuthGroupName	string	Name of the Active Directory or Windows Group. Max Length=450
Description	string	User description. Default=”” Max Length=250
IsReadOnly	SwitchParameter	Indicates whether the administrative role is read-only(true) or read-write(false). Default=false

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.

Examples

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.

Name	Type	Description
Enabled	uint	1 if CEIP is enabled, otherwise 0. Min=0, Max=1
NextUpload	DateTime	Date and time next CEIP upload is due if enabled is 1. Default=Empty
InProgress	uint	1 if an upload is currently in progress, otherwise 0. Default=0
ServerId	Guid	ID of server that is currently uploading, null if InProgress is 0. Default=00000000-0000-0000-0000-000000000000
OneTimeUpload	uint	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.

Examples

EXAMPLE 1: Create PvsCeipData with Minimum Fields Creates a PvsCeipData with enabled as false.

```
1 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.

```
1 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.

Name	Type	Description
UserName	string	Username used to obtain the token Max Length=255
Path	string	Path where the last problem report bundle was saved Default=""Max Length=255
Password	string	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.

Examples

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.

Name	Type	Description
Name or CollectionName	string	Name of the Collection. It is unique within the Site. Max Length=50
SiteId	Guid	GUID of the Site that this Collection is a member of. It is not used with SiteName.
SiteName	string	Name of the Site that this Collection is a member of. It is not used with SiteId.
Description	string	User description. Default=’”’ Max Length=250
LastAutoAddDeviceNumber	uint	The Device Number of the last Auto Added Device. Default=0
Disabled	SwitchParameter	If -Disabled is specified, the Devices in the Collection can not be booted. By default the Devices can be booted.
AutoAddPrefix	string	The string put before the Device Number for Auto Add. Default=’”’ASCII computer name characters no end digit Max Length=12

Name	Type	Description
AutoAddSuffix	string	The string put after the Device Number for Auto Add. Default=””ASCII computer name characters no begin digit Max Length=12
NoAutoAddZeroFill	SwitchParameter	If -NoAutoAddZeroFill is specified, zeros will not be placed before the Device Number up to the AutoAddNumberLength for Auto Add.
AutoAddNumberLength	uint	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 ((lenautoAddPre-fix+lenautoAddSuffix)+AutoAddNumberLeng Min=3, Max=9, Default=4
ProvisioningType	uint	The provisioning facility which created the Devices in this Collection. 0 is PVS, 1 is Studio. Only writable when adding a colection. Min=0, Max=1, 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.
PvsCollection: If successful, the new PvsCollection object is returned.

Examples

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.

Name	Type	Description
Name or DeviceName	string	Computer name with no spaces. ASCII computer name characters Max Length=15
CollectionId	Guid	GUID of the Collection this Device is to be a member of. It is not used with CollectionName.
CollectionName	string	Name of the Collection this Device is to be a member of. SiteName or SiteId must also be used.
SiteId	Guid	GUID of the Site the CollectionName is to be a member of. This or SiteName is used with CollectionName.

Name	Type	Description
SiteName	string	Name of the Site the CollectionName is to be a member of. This or SiteId is used with CollectionName.
Description	string	User description. Default=’”’ Max Length=250
DeviceMac	PvsPhysicalAddress	Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device. Default=’”’
BootFrom	uint	Device to boot from. Choices are 1 for vDisk, 2 for Hard Disk, and 3 for Floppy. Min=1, Max=3, Default=1
ClassName	string	Used by Automatic Update feature to match new versions of Disks to a Device. Default=’”’ Max Length=41
Port	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901
Disabled	SwitchParameter	If -Disabled is specified, the Device can not be booted. By default the Device can be booted.
LocalDiskEnabled	SwitchParameter	If -LocalDiskEnabled is specified, there will be a local disk menu choice for the Device.
Authentication	uint	Device log in authentication. Choices are 0 for none, 1 for User Name/Password, and 2 for Extern. Min=0, Max=2, Default=0
User	string	Name of user to authenticate before the boot process continues. Default=’”’ASCII Max Length=20

Name	Type	Description
Password	string	Password of user to authenticate before the boot process continues. Default=”” ASCII Max Length=65535
CopyTemplate	SwitchParameter	If -CopyTemplate is specified, the Template Device for the collection, if it exists, will be used for the property settings of the new Device.
AdTimestamp	uint	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	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0
AdPassword	string	The Active Directory machine account password. Default=”” ASCII Max Length=65535
LogLevel	uint	Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
DomainName	string	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	string	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

Name	Type	Description
DomainControllerName	string	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	DateTime	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	uint	1 when it performs test of Disks, 2 when it performs maintenance on Disks, 0 otherwise. Min=0, Max=2, Default=0
LocalWriteCacheDiskSize	uint	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	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	string	Hypervisor VM ID for HCL Default=''Max Length=65535
BdmBoot	SwitchParameter	If -BdmBoot is 0, use PXE, 1 use BDM. PXE boot is used by default.
BdmType	uint	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	uint	1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0

Name	Type	Description
BdmUpdated	DateTime	Timestamp of the last BDM boot disk update. Default=Empty
BdmCreated	DateTime	Timestamp when BDM device was created Default=Empty
XsPvsProxyUuid	Guid	UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000
EnableXsProxy	string	Enable XenServerProxy when set to 1 Default=’”’
WriteCacheDisk	string	Write cache disk number Default=’”’Max Length=3
WriteCachePartition	string	Write cache disk partition number Default=’”’Max Length=3
UpdatePageFileSettings	SwitchParameter	Update pagefile settings Default=false
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0
ProvisioningType	uint	The provisioning facility which created the device. 0 is PVS, 1 is Studio. Only writable when adding a device. Min=0, Max=1, Default=0
EkPub	string	Client TPM EK public key Default=’”’
SecurityPolicy	uint	Client security policy Expected values are 0, 2 or 10, Default=null.

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.

Examples

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 "15
    e0544e-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.

Name	Type	Description
Name or DeviceName	string	Computer name with no spaces. ASCII computer name characters Max Length=15
CollectionId	Guid	GUID of the Collection this Device with Personal vDisk is to be a member of. It is not used with CollectionName.
CollectionName	string	Name of the Collection this Device with Personal vDisk is to be a member of. SiteName or SiteId must also be used.
DiskLocatorId	Guid	GUID of the Disk Locator to update with this Device.
SiteId	Guid	GUID of the Site the CollectionName is to be a member of. This or SiteName is used with CollectionName.
SiteName	string	Name of the Site the CollectionName is to be a member of. This or SiteId is used with CollectionName.
Description	string	User description. Default=”” Max Length=250
DeviceMac	PvsPhysicalAddress	Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device with Personal vDisk.
Port	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901
AdTimestamp	uint	The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

Name	Type	Description
AdSignature	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0
LogLevel	uint	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
DomainName	string	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
DomainObjectSID	string	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
DomainControllerName	string	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	DateTime	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

Name	Type	Description
Type	uint	3 when it has a Personal vDisk, 4 when it has a Personal vDisk and performs tests. Min=3, Max=4, Default=3
PvdDriveLetter	string	Personal vDisk Drive letter. Range is F to Z. Default=’”’Max Length=1
LocalWriteCacheDiskSize	uint	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
BdmBoot	SwitchParameter	If -BdmBoot is specified, user BDM instead of PXE boot. PXE boot is used by default.
XdSiteId	Guid	GUID of the XenDesktop Site. Default=00000000-0000-0000-0000-000000000000
XdCatalogId	uint	Integer identifier of the XenDesktop Catalog. Default=””
VirtualHostingPoolId	Guid	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
EnableXsProxy	string	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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to create a Directory on.
Name or ServerName	string[]	Name of the Server to create a Directory on.

This is required

Name	Type	Description
Path	string[]	

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 pipeline:
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.

Examples

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.

Name	Type	Description
Name or DiskLocatorName	string	Name of the Disk Locator File. It is unique within the Store. ASCII Max Length=52
SiteId	Guid	GUID of the Site this DiskLocator is to be a member of. It is not used with SiteName.
SiteName	string	Name of the Site this DiskLocator is to be a member of. It is not used with SiteId.
StoreId	Guid	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	string	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	string	User description. Default=’”’ Max Length=250
MenuText	string	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	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	string	Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=’”’

Name	Type	Description
Disabled	SwitchParameter	If -Disabled is specified, the disk can not be booted. By default the disk can be booted.
RebalanceEnabled	SwitchParameter	If -RebalanceEnabled is specified, this Server can automatically rebalance Devices.
RebalanceTriggerPercent	uint	Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25
SubnetAffinity	uint	Qualifier for subnet affinity when assigning a Server. 0=None, 1=Best Effort, 2=Fixed. Min=0, Max=2, Default=0
NewDiskWriteCacheType	string	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
VHDX	SwitchParameter	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.

Examples

EXAMPLE 1: Create PvsDiskLocator with Minimum Fields Creates a PvsDiskLocator named theDiskLocator in theSite and theStore.

```
1 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.

```
1 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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid	GUID of the Disk Locator File to Enable Disk Maintenance on.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:
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.

Examples

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.

Name	Type	Description
Name or DeviceName	string	Computer name with no spaces. ASCII computer name characters Max Length=15
VirtualHostingPoolId	Guid	GUID of the Virtual Hosting Pool. It is not used with VirtualHostingPoolName.
VirtualHostingPoolName	string	Name of the Virtual Hosting Pool.
DiskLocatorId	Guid	GUID of the Disk Locator to update with this Device.
Description	string	User description. Default=’”’ Max Length=250
DeviceMac	PvsPhysicalAddress	Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device. Default=’”’
Port	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901
AdTimestamp	uint	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	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0
LogLevel	uint	Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4

Name	Type	Description
DomainName	string	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	string	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	string	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	DateTime	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	string	The Active Directory machine account password. Default=’”’ Max Length=65535

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.

Examples

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 "2888
    f183-2f48-4771-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 "2888
    f183-2f48-4771-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.

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

Examples

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.

```
1 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.

Name	Type	Description
Name or ServerName	string	Computer name with no spaces. ASCII computer name characters Max Length=21
SiteId	Guid	GUID of the Site this Server is to be a member of. It is not used with SiteName.
SiteName	string	Name of the Site this Server is to be a member of. It is not used with SiteId.
Description	string	User description. Default=”” Max Length=250
AdMaxPasswordAge	uint	Number of days before a password expires. Min=1, Max=30, Default=7
LicenseTimeout	uint	Amount of seconds before a license times out. Min=15, Max=300, Default=30
VDiskCreatePacing	uint	VDisk create time pacing in milliseconds. Min=0, Max=5, Default=0
FirstPort	uint	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

Name	Type	Description
LastPort	uint	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	uint	Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8
BuffersPerThread	uint	Number of buffers per worker thread. Min=1, Max=128, Default=24
ServerCacheTimeout	uint	Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8
IoBurstSize	uint	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	uint	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	uint	Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500
MaxBootSeconds	uint	Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60

Name	Type	Description
BootPauseSeconds	uint	Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10
AdMaxPasswordAgeEnabled	SwitchParameter	If -AdMaxPasswordAgeEnabled is specified, Age the password.
EventLoggingEnabled	SwitchParameter	If -EventLoggingEnabled is specified, event logging is enabled.
NonBlockingIoDisabled	SwitchParameter	If -NonBlockingIoDisabled is specified, do not use non-Blocking IO.
Ip	string[]	One or more streaming IPv4 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv4 addresses
Ipv6	string[]	One or more streaming IPv6 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv6 addresses
InitialQueryConnectionPoolSize	uint	Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50
InitialTransactionConnectionPoolSize	uint	Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50
MaxQueryConnectionPoolSize	uint	Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000

Name	Type	Description
MaxTransactionConnectionPoolSize	uint	Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000
RefreshInterval	uint	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	uint	Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300
BusyDbConnectionRetryCount	uint	Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2
BusyDbConnectionRetryInterval	uint	Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25
LocalConcurrentIoLimit	uint	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	uint	Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4

Name	Type	Description
LogLevel	uint	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	uint	Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5
LogFileBackupCopiesMax	uint	Maximum number of log file backups. Min=1, Max=50, Default=4
PowerRating	float	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
LastCeipUploadAttempt	DateTime	Time that this server last attempted a CEIP upload. Default=Empty
LastBugReportAttempt	DateTime	Time that this server last attempted to upload or generate a bug report bundle. Default=Empty
LastBugReportStatus	string	Status of the last bug report on this server. Default='"'Max Length=250
LastBugReportResult	string	Status of the last bug report on this server. Default='"'Max Length=4000
LastBugReportSummary	string	Summary of the last bug report on this server. Default='"'Max Length=250

Name	Type	Description
ServerEpoch	UInt64	Timestamp when the registry data for that server has been re-encrypted with updated encryption scheme. Default=0
SecurityPolicy	uint	SecurityPolicy Default=0
QuicSettings	string	QuicSettings Default=’”’Max Length=65535
QuicPort	uint	QuicPort Default=0
QuicServerCertThumbPrint	string	QuicServerCertThumbPrint Default=’”’Max Length=65535
QuicCipherSuites	uint	QuicCipherSuites Default=’”’
QuicStreamPayloadLength	uint	QuicStreamPayloadLength Default=0
TpmKeyType	uint	TPM key type, TPM_ALG_RSA(1), TPM_ALG_ECC(0x23) Default=0
TpmHashAlg	uint	TPM hash algorithm, TPM_ALG_SHA256(0xb), TPM_ALG_SHA384(0xc), TPM_ALG_SHA512(0xd) Default=0
TpmCurveId	uint	TPM ECC curve ID TPM_ECC_NIST_P256(3), TPM_ECC_NIST_P384(4), TPM_ECC_NIST_P521(5) 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.
PvsServer: If successful, the new PvsServer object is returned.

Examples

EXAMPLE 1: Create PvsServer with Minimum Fields Creates a PvsServer named theServer in the-Site.


```
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.

Name	Type	Description
Name or SiteName	string	Name of the Site. Max Length=50
Description	string	User description. Default=”” Max Length=250
InventoryFilePollingInterval	uint	The number of seconds between polls for Disk changes in the Stores. Min=1, Max=600, Default=60
EnableDiskUpdate	SwitchParameter	If -EnableDiskUpdate is specified, the Disk Update will be enabled for the Site. By default Disk Update is disabled.

Name	Type	Description
DiskUpdateServerId	Guid	GUID of the Disk Update Server for the Site. Not used with DiskUpdateServerName. Default=00000000-0000-0000-0000-000000000000
DiskUpdateServerName	string	Name of the Disk Update Server for the Site. Not used with DiskUpdateServerId. Default=””
MakUser	string	User name used for MAK activation. Default=””Max Length=64
MakPassword	string	User password used for MAK activation. Default=””Max Length=64
EnableXsProxy	string	Enable XenServerProxy when set to 1 Default=””
VirtualHostingPoolId	Guid	GUID of the VirtualHostingPool object.
VirtualHostingPoolName	string	Name of the VirtualHostingPool object.
XsPvsSiteUuid	string	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.

Examples

EXAMPLE 1: Create PvsSite with Minimum Fields Creates a PvsSite named theSite.

```
1  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.

Name	Type	Description
Name or SiteViewName	string	Name of the Site View. Max Length=50
SiteId	Guid	GUID of the Site this View is to be a member of. It is not used with SiteName.
SiteName	string	Name of the Site this View is to be a member of. It is not used with SiteId.
Description	string	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.

Examples

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 the-Site with “XenServer SiteView”as the description.

```
1 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.

Name	Type	Description
Name or StoreName	string	Name of the Store. Max Length=50
SiteId	Guid	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	string	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	string	User description. Default=”” Max Length=250
Path	string	Default directory path that the Servers use to access this Store. Max Length=255
CachePath	string[]	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.

Examples

EXAMPLE 1: Create PvsStore with Minimum Fields Creates a PvsStore named theStore with network share path.

```
1 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.

```
1 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.

Name	Type	Description
Name or UpdateTaskName	string	Name of the Update Task. It is unique within the Site. Max Length=50
SiteId	Guid	GUID of the Site that this Update Task is a member of. It is not used with SiteName.
SiteName	string	Name of the Site that this Update Task is a member of. It is not used with SiteId.
Description	string	User description. Default=”” Max Length=250
Disabled	SwitchParameter	If -Disabled is specified, the updates will not be processed. By default the updates will be processed.

Name	Type	Description
Hour	uint	The hour of the day to perform the task. Min=0, Max=23, Default=0
Minute	uint	The minute of the hour to perform the task. Min=0, Max=59, Default=0
Recurrence	uint	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	uint	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	uint[]	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	uint	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	string	Esd to use. Valid values are SCCM or WSUS. If no value, a custom script is run on the client. Default=’”’Max Length=50

Name	Type	Description
PreUpdateScript	string	Script file to run before the update starts. Default=’”’Max Length=255
PreVmScript	string	Script file to run before the VM is loaded. Default=’”’Max Length=255
PostUpdateScript	string	Script file to run after the update finishes. Default=’”’Max Length=255
PostVmScript	string	Script file to run after the VM is unloaded. Default=’”’Max Length=255
Domain	string	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

Name	Type	Description
OrganizationUnit	string	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	uint	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.

Examples

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.

Name	Type	Description
Name or VirtualHostingPoolName	string	Name of the Virtual Hosting Pool. It is unique within the Site. Max Length=50
SiteId	Guid	GUID of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteName.
SiteName	string	Name of the Site that this Virtual Hosting Pool is a member of. It is not used with SiteId.
Type	uint	Type of the Virtual Hosting Pool. 0 = Citrix XenServer, 1 = Microsoft SCVMM/Hyper-V, 2 = VMWare vSphere/ESX, 3 = Nutanix, 4 = Azure, 5 = GCP. Min=0, Max=255, Default=0
Description	string	User description. Default=’”’ Max Length=250
Server	string	Name or IP of the Host Server. Max Length=255
Port	uint	Port of the Host Server. Min=80, Max=65534, Default=80
Datacenter	string	Datacenter of the Virtual Hosting Pool. Default=’”’Max Length=250
UpdateLimit	uint	Number of updates at the same time. Min=2, Max=1000, Default=1000
UpdateTimeout	uint	Timeout for updates. Min=2, Max=240, Default=60
ShutdownTimeout	uint	Timeout for shutdown. Min=2, Max=30, Default=10

Name	Type	Description
UserName	string	Name to use when logging into the Server.
Password	string	Password to use when logging into the Server.
XdHostingUnitUuid	Guid	UUID of XenDesktop Hosting Unit Default=00000000-0000-0000-0000-000000000000
PrepopulateEnabled	SwitchParameter	Enable prepopulate when set to true Default=false
XsPvsSiteUuid	Guid	UUID of XenServer PVS_site Default=00000000-0000-0000-0000-000000000000
PlatformVersion	string	Hypervisor Host Version Default=’”’Max Length=250
XdHcHypervisorConnectionName	string	Hypervisor Connection Name for HCL Connection Details object Default=’”’Max Length=250
XdHcHypervisorConnectionUid	string	Hypervisor Connection Uid for HCL Connection Details object Default=’”’Max Length=250
XdHcRevision	string	Revision for HCL Connection Details object Default=’”’Max Length=250
XdHcCustomProperties	string	Custom Properties for HCL Connection Details object Default=’”’Max Length=65535
XdHcSslThumbprints	string	Ssl Thumbprints for HCL Connection Details object Default=’”’Max Length=65535
CredentialsUid	string	UUID of any record associated with the credentials Default=’”’ Max Length=65535
DdcType	uint	Type of the DDC. 0 = Unset, 1 = Citrix Cloud, 2 = customer-managed. Min=0, Max=2, Default=0

Name	Type	Description
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0
DisableHostXsProxy	string	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.

Examples

EXAMPLE 1: Create PvsVirtualHostingPool with Minimum Fields Creates a PvsVirtualHosting-Pool named theVirtualHostingPool in theSite.

```
1  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.

```
1  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 is required

Name	Type	Description
Guid or AuthGroupId	Guid[]	GUID of the AuthGroup to Delete.
Name or AuthGroupName	string[]	Name of the AuthGroup to Delete.

Optional

Name	Type	Description
Force	SwitchParameter	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or CollectionId	Guid[]	GUID of the Collection to Delete.

or this is required & resolution

Name	Type	Description
Name or CollectionName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Delete.
Name or DeviceName	string[]	Name of the Device to Delete.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Delete.
EkPub	string[]	TPM EK public key of the Device to Delete
CollectionId	Guid[]	GUID of the Collection to delete all Devices.

or this is required & resolution

Name	Type	Description
CollectionName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Delete Disk cache files.
Name or DeviceName	string[]	Name of the Device to Delete Disk cache files.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Delete Disk cache files.

This is required

Name	Type	Description
DiskLocatorId	Guid[]	

or this is required & resolution

Name	Type	Description
DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Remove from the Domain.
Name or DeviceName	string[]	Name of the Device to Remove from the Domain.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Remove from the Domain.
CollectionId	Guid[]	GUID of the Collection to Remove all Devices from the Domain.
SiteViewId	Guid[]	GUID of the Site View to Remove all Devices from the Domain.
FarmViewId	Guid[]	GUID of the Farm View to Remove all Devices from the Domain.
FarmViewName	string[]	Name of the Farm View toRemove all Devices from the Domain.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string[]	
SiteViewName	string[]	

Optional

Name	Type	Description
Domain	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Remove.
Name or DeviceName	string[]	Name of the Device to Remove.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Remove.

One of these is required

Name	Type	Description
SiteViewId	Guid[]	
FarmViewId	Guid[]	
FarmViewName	string[]	

or this is required & resolution

Name	Type	Description
SiteViewName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to remove a Directory from. The directory must be empty to be removed.
Name or ServerName	string[]	Name of the Server to remove a Directory from. The directory must be empty to be removed.

This is required

Name	Type	Description
Path	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Remove.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

These are optional

Name	Type	Description
UpdateTaskId	Guid[]	
UpdateTaskName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Delete.

or one of these is required & resolutions

Name	Type	Description
Name or DiskLocatorName	string[]	

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

Optional

Name	Type	Description
DeleteDiskFile	SwitchParameter	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Remove.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

These are optional

Name	Type	Description
DeviceId	Guid[]	
DeviceName	string[]	
DeviceMac	PvsPhysicalAddress[]	
CollectionId	Guid[]	
SiteViewId	Guid[]	
FarmViewId	Guid[]	
FarmViewName	string[]	

These are optional & resolutions

Name	Type	Description
CollectionName	string[]	
SiteViewName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Disk Update Device to Delete.
Name or DeviceName	string[]	Name of the Disk Update Device to Delete.
DeviceMac	PvsPhysicalAddress[]	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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to Delete the Version from.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

Optional

Name	Type	Description
Version	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
(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.

Examples

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 is required

Name	Type	Description
Guid or FarmViewId	Guid[]	GUID of the Farm View to Delete.
Name or FarmViewName	string[]	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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to Delete.
Name or ServerName	string[]	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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of a Server that uses the path to get to the Store.
Name or ServerName	string[]	Name of a Server that uses the path to get to the Store.

One of these is required

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

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.

Examples

EXAMPLE 1: Remove the PvsServerStore with Name theServer/theStore This example removes the PvsServerStore for the Server named theServer and Store named theStore.

```
1  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.

```
1  Get-PvsServer -Name theServer | Remove-PvsServerStore -StoreName
       theStore
```


Remove-PvsSite

Remove one or more Sites.

One of these is required

Name	Type	Description
Guid or SiteId	Guid[]	GUID of the Site to Delete.
Name or SiteName	string[]	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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or SiteViewId	Guid[]	GUID of the Site View to Delete.

or this is required & resolution

Name	Type	Description
Name or SiteViewName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or StoreId	Guid[]	GUID of the Store to Delete.
Name or StoreName	string[]	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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or UpdateTaskId	Guid[]	GUID of the Update Task to Delete.

or this is required & resolution

Name	Type	Description
Name or UpdateTaskName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or VirtualHostingPoolId	Guid[]	GUID of the Virtual Hosting Pool to Delete.

or this is required & resolution

Name	Type	Description
Name or VirtualHostingPoolName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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.

Examples

EXAMPLE 1: Reset-PvsDatabase

```
Reset-PvsDatabase
```

Reset-PvsDeviceForDomain

Reset a Device, all Devices in a Collection or View for a Domain.

One of these is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Reset for the Domain.
Name or DeviceName	string[]	Name of the Device to Reset for the Domain.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Reset for the Domain.

Name	Type	Description
CollectionId	Guid[]	GUID of the Collection to Reset all Devices for the Domain.
SiteViewId	Guid[]	GUID of the Site View to Reset all Devices for the Domain.
FarmViewId	Guid[]	GUID of the Farm View to Reset all Devices for the Domain.
FarmViewName	string[]	Name of the Farm View to Reset all Devices for the Domain.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string[]	
SiteViewName	string[]	

Optional

Name	Type	Description
Domain	string[]	
OrganizationUnit	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to restart the Stream Service.
Name or ServerName	string[]	Name of the Server to restart the Stream Service.
SiteId	Guid[]	GUID of the Site to restart the Stream Service on all Servers.
SiteName	string[]	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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or AuthGroupId	Guid[]	GUID of the AuthGroup to remove Authorization for.
Name or AuthGroupName	string[]	Name of the AuthGroup to remove Authorization for.

These are optional

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	
CollectionId	Guid[]	

or this is optional & resolution

Name	Type	Description
CollectionName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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

Name	Type	Description
Name or AuthGroupName	string	Name of the Active Directory or Windows Group. Max Length=450
Description	string	User description. Default=’”’ Max Length=250
IsReadOnly	bool	Indicates whether the administrative role is read-only(true) or read-write(false). Default=false

When AuthGroup is not passed, the parameters below are used:
One of these is required

Name	Type	Description
Guid or AuthGroupId	Guid	GUID of the AuthGroup to Set.
Name or AuthGroupName	string	Name of the AuthGroup to Set.

Optional field values to set:

Name	Type	Description
NewName	string	Name of the Active Directory or Windows Group. Max Length=450
Description	string	User description. Default=”” Max Length=250
IsReadOnly	bool	Indicates whether the administrative role is read-only(true) or read-write(false). Default=false

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.

Examples

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

Name	Type	Description
Enabled	uint	1 if CEIP is enabled, otherwise 0. Min=0, Max=1
NextUpload	DateTime	Date and time next CEIP upload is due if enabled is 1. Default=Empty
InProgress	uint	1 if an upload is currently in progress, otherwise 0. Default=0
ServerId	Guid	ID of server that is currently uploading, null if InProgress is 0. Default=00000000-0000-0000-0000-000000000000
OneTimeUpload	uint	1 to perform a one time upload. Default=0

When CeipData is not passed, the parameters below are used:
Optional

Name	Type	Description
Uuid	string	CEIP UUID of this Farm. This is optional since there is only one.

Optional field values to set:

Name	Type	Description
Enabled	uint	1 if CEIP is enabled, otherwise 0. Min=0, Max=1

Name	Type	Description
NextUpload	DateTime	Date and time next CEIP upload is due if enabled is 1. Default=Empty
InProgress	uint	1 if an upload is currently in progress, otherwise 0. Default=0
ServerId	Guid	ID of server that is currently uploading, null if InProgress is 0. Default=00000000-0000-0000-0000-000000000000
OneTimeUpload	uint	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.

Examples

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  }
```



```
5    | 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 pipeline.
These values are in the PvsCisData object, and only will be set if the value has changed.

Name	Type	Description
UserName	string	Username used to obtain the token Default=’”’Max Length=255
Path	string	Path where the last problem report bundle was saved Default=’”’Max Length=255
Password	string	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

Name	Type	Description
Guid or CisDataId	Guid	CIS UUID of this Farm. This is optional since there is only a single record.

Optional field values to set:

Name	Type	Description
UserName	string	Username used to obtain the token Default=””Max Length=255
Path	string	Path where the last problem report bundle was saved Default=””Max Length=255
Password	string	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.

Examples

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

Name	Type	Description
Name or CollectionName	string	Name of the Collection. It is unique within the Site. Max Length=50
Description	string	User description. Default=’”’ Max Length=250
TemplateDeviceId	Guid	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	string	Name of a Device in the Collection whose settings are used for initial values of new Devices. Not used with TemplateDeviceId. Default=’”’
LastAutoAddDeviceNumber	uint	The Device Number of the last Auto Added Device. Default=0
Enabled	bool	True when Devices in the Collection can be booted, false otherwise. Default=true

Name	Type	Description
AutoAddPrefix	string	The string put before the Device Number for Auto Add. Default=’”’ASCII computer name characters no end digit Max Length=12
AutoAddSuffix	string	The string put after the Device Number for Auto Add. Default=’”’ASCII computer name characters no begin digit Max Length=12
AutoAddZeroFill	bool	True when zeros be placed before the Device Number up to the AutoAddNumberLength for Auto Add, false otherwise. Default=true
AutoAddNumberLength	uint	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 ((lenautoAddPre-fix+lenautoAddSuffix)+AutoAddNumberLeng Min=3, Max=9, Default=4
ProvisioningType	uint	The provisioning facility which created the Devices in this Collection. 0 is PVS, 1 is Studio. Only writable when adding a colection. Min=0, Max=1, Default=0

When Collection is not passed, the parameters below are used:
This is required

Name	Type	Description
Guid or CollectionId	Guid	GUID of the Collection to Set.

or this is required & resolution

Name	Type	Description
Name or CollectionName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

Optional field values to set:

Name	Type	Description
NewName	string	Name of the Collection. It is unique within the Site. Max Length=50
Description	string	User description. Default=’”’ Max Length=250
TemplateDeviceId	Guid	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	string	Name of a Device in the Collection whose settings are used for initial values of new Devices. Not used with TemplateDeviceId. Default=’”’
LastAutoAddDeviceNumber	uint	The Device Number of the last Auto Added Device. Default=0
Enabled	bool	True when Devices in the Collection can be booted, false otherwise. Default=true

Name	Type	Description
AutoAddPrefix	string	The string put before the Device Number for Auto Add. Default=””ASCII computer name characters no end digit Max Length=12
AutoAddSuffix	string	The string put after the Device Number for Auto Add. Default=””ASCII computer name characters no begin digit Max Length=12
AutoAddZeroFill	bool	True when zeros be placed before the Device Number up to the AutoAddNumberLength for Auto Add, false otherwise. Default=true
AutoAddNumberLength	uint	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 ((lenautoAddPre-fix+lenautoAddSuffix)+AutoAddNumberLeng Min=3, Max=9, Default=4
ProvisioningType	uint	The provisioning facility which created the Devices in this Collection. 0 is PVS, 1 is Studio. Only writable when adding a colection. Min=0, Max=1, Default=0

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.

Examples

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

Name	Type	Description
Name or Server	string	Name or IP of the Server to connect to. Default=localhost
Port	string	The Port to use to connect. Default=54321
User	string	User name to use for Authentication. If it has a value, it will be *****. Default=’”’
Domain	string	Domain name to use for Authentication. If it has a value, it will be *****. Default=’”’
Password	string	Password to use for Authentication. If it has a value, it will be *****. Default=’”’
Persist	string	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:

Name	Type	Description
Name or Server	string	Name or IP of the Server to connect to. Default=localhost
Port	string	The Port to use to connect. Default=54321
User	string	User name to use for Authentication. If it has a value, it will be *****. Default=’”’
Domain	string	Domain name to use for Authentication. If it has a value, it will be *****. Default=’”’
Password	string	Password to use for Authentication. If it has a value, it will be *****. Default=’”’
Persist	string	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.

Examples

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

Name	Type	Description
Name or DeviceName	string	Computer name with no spaces. ASCII computer name characters Max Length=15
Description	string	User description. Default="" Max Length=250
DeviceMac	PvsPhysicalAddress	Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device. Default=""

Name	Type	Description
BootFrom	uint	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	string	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	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901
Enabled	bool	True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true
LocalDiskEnabled	bool	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
Authentication	uint	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	string	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

Name	Type	Description
Password	string	Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default=’”’ASCII Max Length=65535
AdTimestamp	uint	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	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0
AdPassword	string	The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=’”’ASCII Max Length=65535
LogLevel	uint	Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
DomainName	string	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	string	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

Name	Type	Description
DomainControllerName	string	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	DateTime	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	uint	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
LocalWriteCacheDiskSize	uint	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
BdmBoot	bool	Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false
BdmType	uint	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	uint	1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0

Name	Type	Description
BdmUpdated	DateTime	Timestamp of the last BDM boot disk update. Default=Empty
BdmCreated	DateTime	Timestamp when BDM device was created Default=Empty
XsPvsProxyUuid	Guid	UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000
EnableXsProxy	string	Enable XenServerProxy when set to 1 Default=’”’
WriteCacheDisk	string	Write cache disk number Default=’”’Max Length=3
WriteCachePartition	string	Write cache disk partition number Default=’”’Max Length=3
UpdatePageFileSettings	bool	Update pagefile settings Default=false
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0
ProvisioningType	uint	The provisioning facility which created the device. 0 is PVS, 1 is Studio. Only writable when adding a device. Min=0, Max=1, Default=0
EkPub	string	Client TPM EK public key Default=’”’
SecurityPolicy	uint	Client security policy Expected values are 0, 2 or 10, Default=null.

When Device is not passed, the parameters below are used:
One of these is required

Name	Type	Description
Guid or DeviceId	Guid	GUID of the Device to Set.

Name	Type	Description
Name or DeviceName	string	Name of the Device to Set.
DeviceMac	PvsPhysicalAddress	MAC of the Device to Set.
CollectionId	Guid	GUID of the Collection to set all Devices. DeviceName and DeviceMac cannot be set.
SiteViewId	Guid	GUID of the Site View to set all Devices. DeviceName and DeviceMac cannot be set.
FarmViewId	Guid	GUID of the Farm View to set all Devices. DeviceName and DeviceMac cannot be set.
FarmViewName	string	Name of the Farm View to set all Devices. DeviceName and DeviceMac cannot be set.
EkPub	string	TPM endorsement public key of device to set.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

Optional field values to set:

Name	Type	Description
NewName	string	Computer name with no spaces. ASCII computer name characters Max Length=15
Description	string	User description. Default=’”’ Max Length=250
DeviceMac	PvsPhysicalAddress	Ethernet address can have the form XX-XX-XX-XX-XX-XX. Uniquely identifies the Device. Default=’”’
BootFrom	uint	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	string	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	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901
Enabled	bool	True when it can be booted, false otherwise. This cannot be Set for a Device with Personal vDisk. Default=true
LocalDiskEnabled	bool	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

Name	Type	Description
Authentication	uint	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	string	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	string	Password of user to authenticate before the boot process continues. This cannot be Set for a Device with Personal vDisk. Default=’”’ASCII Max Length=65535
AdTimestamp	uint	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	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0
AdPassword	string	The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=’”’ASCII Max Length=65535
LogLevel	uint	Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4

Name	Type	Description
DomainName	string	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	string	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	string	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	DateTime	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	uint	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

Name	Type	Description
LocalWriteCacheDiskSize	uint	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
BdmBoot	bool	Use PXE boot when set to false, BDM boot when set to true. Default is PXE Default=false
BdmType	uint	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	uint	1 use VHD for BDMboot, 2 use ISO, 3 use USB. Default=0
BdmUpdated	DateTime	Timestamp of the last BDM boot disk update. Default=Empty
BdmCreated	DateTime	Timestamp when BDM device was created Default=Empty
XsPvsProxyUuid	Guid	UUID of XenServer PVS_proxy Default=00000000-0000-0000-0000-000000000000
EnableXsProxy	string	Enable XenServerProxy when set to 1 Default=’”’
WriteCacheDisk	string	Write cache disk number Default=’”’Max Length=3
WriteCachePartition	string	Write cache disk partition number Default=’”’Max Length=3
UpdatePageFileSettings	bool	Update pagefile settings Default=false
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0

Name	Type	Description
ProvisioningType	uint	The provisioning facility which created the device. 0 is PVS, 1 is Studio. Only writable when adding a device. Min=0, Max=1, Default=0
EkPub	string	Client TPM EK public key Default=””
SecurityPolicy	uint	Client security policy Expected values are 0, 2 or 10, Default=null.

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.

Examples

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.

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

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

Examples

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.

Name	Type	Description
Name	string	Name of the Device personality item. Max Length=250
Value	string	Value for the Device personality item. Max Length=1000

Examples

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

Name	Type	Description
Class	string	Class of the Disk. Max Length=40
ImageType	string	Type of this image (software type). Max Length=40

Name	Type	Description
WriteCacheSize	UInt64	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	bool	Automatically update this image for matching Devices when set to true. Default false
ActivationDateEnabled	bool	Use activation date to activate image when set to true. Default false
AdPasswordEnabled	bool	Enable AD password management when set to true.
HaEnabled	bool	Enable HA when set to true.
PrinterManagementEnabled	bool	Invalid printers will be deleted from the Device when set to true.
WriteCacheType	uint	0 (Private), 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), 9 (Cache in Device RAM with Overflow on Hard Disk), 10 (Private async), 11(Server persistent async), 12 (Cache in Device RAM with Overflow on Hard Disk async) Min=0, Max=12, Default=0
LicenseMode	uint	0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0

Name	Type	Description
AccelerateOfficeActivation	bool	Run script to activate office automatically.
ActiveDate	DateTime	Date to activate the disk if AutoUpdateEnabled and activationDateEnabled are true. Has the date. Empty when the AutoUpdateEnabled or activationDateEnabled are false.
LongDescription	string	Description of the Disk. Max Length=399
OperatingSystem	string	Operating System of Disk. Max Length=250
OsType	string	Operating System Type of Disk. Max Length=40
SerialNumber	string	User defined serial number. Max Length=36
Date	string	User defined date. Max Length=40
Author	string	User defined author. Max Length=40
Title	string	User defined title. Max Length=40
Company	string	User defined company. Max Length=40
InternalName	string	User defined name. Max Length=63
OriginalFile	string	User defined original file. Max Length=127
HardwareTarget	string	User defined hardware target. Max Length=127
MajorRelease	UInt64	User defined major release number. Min=0, Max=4294967295, Default=0
MinorRelease	UInt64	User defined minor release number. Min=0, Max=4294967295, Default=0

Name	Type	Description
Build	UInt64	User defined build number. Min=0, Max=4294967295, Default=0
ClearCacheDisabled	string	Clear cached secrets disabled.
VHDX	bool	If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false
EnableSetLocalTime	string	Enable setting the local time bias on the disk version.

When Disk is not passed, the parameters below are used:
This is required

Name	Type	Description
Guid or DiskLocatorId	Guid	GUID of the Disk Locator.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	

Name	Type	Description
StoreName	string	

Optional field values to set:

Name	Type	Description
Class	string	Class of the Disk. Max Length=40
ImageType	string	Type of this image (software type). Max Length=40
WriteCacheSize	UInt64	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	bool	Automatically update this image for matching Devices when set to true. Default false
ActivationDateEnabled	bool	Use activation date to activate image when set to true. Default false
AdPasswordEnabled	bool	Enable AD password management when set to true.
HaEnabled	bool	Enable HA when set to true.
PrinterManagementEnabled	bool	Invalid printers will be deleted from the Device when set to true.

Name	Type	Description
WriteCacheType	uint	0 (Private), 1 (Cache on Server), 3 (Cache in Device RAM), 4 (Cache on Device Hard Disk), 7 (Cache on Server, Persistent), 9 (Cache in Device RAM with Overflow on Hard Disk), 10 (Private async), 11(Server persistent async), 12 (Cache in Device RAM with Overflow on Hard Disk async) Min=0, Max=12, Default=0
LicenseMode	uint	0 (None), 1 (Multiple Activation Key), or 2 (Key Management Service). Min=0, Max=2, Default=0
AccelerateOfficeActivation	bool	Run script to activate office automatically.
ActiveDate	DateTime	Date to activate the disk if AutoUpdateEnabled and activationDateEnabled are true. Has the date. Empty when the AutoUpdateEnabled or activationDateEnabled are false.
LongDescription	string	Description of the Disk. Max Length=399
OperatingSystem	string	Operating System of Disk. Max Length=250
OsType	string	Operating System Type of Disk. Max Length=40
SerialNumber	string	User defined serial number. Max Length=36
Date	string	User defined date. Max Length=40
Author	string	User defined author. Max Length=40

Name	Type	Description
Title	string	User defined title. Max Length=40
Company	string	User defined company. Max Length=40
InternalName	string	User defined name. Max Length=63
OriginalFile	string	User defined original file. Max Length=127
HardwareTarget	string	User defined hardware target. Max Length=127
MajorRelease	UInt64	User defined major release number. Min=0, Max=4294967295, Default=0
MinorRelease	UInt64	User defined minor release number. Min=0, Max=4294967295, Default=0
Build	UInt64	User defined build number. Min=0, Max=4294967295, Default=0
ClearCacheDisabled	string	Clear cached secrets disabled.
VHDX	bool	If VHDX is true, the format of the image is VHDX. Otherwise it is VHD. Default=false
EnableSetLocalTime	string	Enable setting the local time bias on the disk version.

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.

Examples

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

Name	Type	Description
Description	string	User description. Default=’”’ Max Length=250
MenuText	string	Text that is displayed in the Boot Menu. If this field has no value, the name value is used. Default=’”’ASCII Max Length=64

Name	Type	Description
ServerId	Guid	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	string	Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=’”’
Enabled	bool	True when this disk can be booted, false otherwise. Default=true
RebalanceEnabled	bool	True when this Server can automatically rebalance Devices, false otherwise. Default=false
RebalanceTriggerPercent	uint	Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25
SubnetAffinity	uint	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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid	GUID of the Disk Locator to Set.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

Optional field values to set:

Name	Type	Description
Description	string	User description. Default=’”’ Max Length=250
MenuText	string	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	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	string	Name of the single Server that this Disk Locator is assigned to. It is not used with ServerId. Default=’”’
Enabled	bool	True when this disk can be booted, false otherwise. Default=true

Name	Type	Description
RebalanceEnabled	bool	True when this Server can automatically rebalance Devices, false otherwise. Default=false
RebalanceTriggerPercent	uint	Percent over fair load that triggers a dynamic Device rebalance. Min=5, Max=5000, Default=25
SubnetAffinity	uint	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.

Examples

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.

```
1 $o = Get-PvsDiskLocator -DiskLocatorId "81ea9077-598d-459f-a443-71
     fabd1840bf" -Fields RebalanceEnabled
2 $o.RebalanceEnabled = $true
3 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.

```
1 Get-PvsDiskLocator -DiskLocatorId \"81ea9077-598d-459f-a443-71
     fabd1840bf\" -Fields RebalanceEnabled | foreach {
2   \$o = \$\_; \$o.RebalanceEnabled = \$true; \$o  }
3   | 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 pipeline.
These values are in the PvsDiskUpdateDevice object, and only will be set if the value has changed.

Name	Type	Description
Description	string	User description. Default=’”’ Max Length=250
Port	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901
AdTimestamp	uint	The time the Active Directory machine account password was generated. Do not set this field, it is only set internally by PVS. Default=0

Name	Type	Description
AdSignature	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0
LogLevel	uint	Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
DomainName	string	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	string	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	string	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	DateTime	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	string	The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=’”’Max Length=65535

When DiskUpdateDevice is not passed, the parameters below are used:
One of these is required

Name	Type	Description
Guid or DeviceId	Guid	GUID of the Disk Update Device to Set.
Name or DeviceName	string	Name of Disk Update Device to Set.
DeviceMac	PvsPhysicalAddress	MAC of Disk Update Device to Set.
SiteId	Guid	GUID of the Site. Can be used alone to Set all Disk Update Devices in the Site.
SiteName	string	Name of the Site. Can be used alone to Set all Disk Update Devices in the Site.
DiskLocatorId	Guid	GUID of the DiskLocator to Set the Disk Update Device for.

or this is required & resolution

Name	Type	Description
DiskLocatorName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	GUID of the Site. Can be used alone to Set all Disk Update Devices in the Site.
SiteName	string	Name of the Site. Can be used alone to Set all Disk Update Devices in the Site.

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

Optional field values to set:

Name	Type	Description
Description	string	User description. Default=’”’ Max Length=250
Port	uint	UDP port to use with Stream Service. Min=1025, Max=65534, Default=6901
AdTimestamp	uint	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	uint	The signature of the Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=0
LogLevel	uint	Level to perform logging at. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
DomainName	string	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

Name	Type	Description
DomainObjectSID	string	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	string	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	DateTime	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	string	The Active Directory machine account password. Do not set this field, it is only set internally by PVS. Default=''Max Length=65535

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.

Examples

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

Name	Type	Description
Description	string	User description. Default=”” Max Length=250
ScheduledDate	DateTime	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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid	GUID of the Disk Locator Version to Set.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string	

This is required

Name	Type	Description
Version	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

Optional field values to set:

Name	Type	Description
Description	string	User description. Default=’”’ Max Length=250

Name	Type	Description
ScheduledDate	DateTime	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.

Examples

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-a443-71
    fabd1840bf" -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-a443-71
    fabd1840bf\" -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 pipeline.
These values are in the PvsFarm object, and only will be set if the value has changed.

Name	Type	Description
Name or FarmName	string	Name of the Farm. Default=’”’ Max Length=50
Description	string	User description. Default=’”’ Max Length=250
AutoAddEnabled	bool	True when Auto Add is enabled, false otherwise. Default=false
AuditingEnabled	bool	True when Auditing is enabled, false otherwise. Default=false
LastAuditArchiveDate	DateTime	Last date of Audit Trail data that was Archived. Has the date. Default=Empty
DefaultSiteId	Guid	GUID of the Site to place new Devices into automatically. Not used with defaultSiteName. Default=00000000-0000-0000-0000-000000000000
DefaultSiteName	string	Name of the Site to place new Devices into automatically. Not used with DefaultSiteId. Default=’”’
OfflineDatabaseSupportEnabled	bool	True when Offline Database Support is enabled, false otherwise. Default=false
LicenseServer	string	License server name. Default=’”’Max Length=255
LicenseServerPort	uint	License server port. Min=1025, Max=65534, Default=27000
LicenseTradeUp	bool	License server trade up, when set to true. Default=false

Name	Type	Description
LicenseSKU	uint	LicenseSKU. 0 for on-premises, 1 for cloud. Min=0, Max=1, Default=0
LicenseWebServicesPort	uint	The license server web services for licensing port. Min=1025, Max=65534, Default=8083
AutomaticMergeEnabled	bool	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	uint	Maximum number a versions of a Disk that can exist before a merge will automatically occur. Min=3, Max=50, Default=5
MergeMode	uint	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
FarmEpoch	UInt64	Timestamp when the Data Encryption Key (DEK) was created and stored in the Credential Wallet. Default=0
EncryptionStatus	uint	The current status for database encryption. 0 = Idle, 1 = Distributing Keys, 2= Re-encrypting database Min=0, Max=2, Default=0
DefaultDeviceLogLevel	uint	The default logging level used by newly created target devices. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4

Name	Type	Description
IsUsageDataScrambled	bool	Indicates whether the sensitive information needs to be scrambled (true) or not (false) when uploading usage telemetry data. Default = true. Default=true
LicenseServerCertThumbprint	string	Stores the license server's server certificate thumbprint when the server certificate is explicitly accepted by the administrator. NULL if no certificate is explicitly accepted. Default=''Max Length=65535

When Farm is not passed, the parameters below are used:
Optional

Name	Type	Description
Guid or FarmId	Guid	GUID of the Farm to Set. This is optional since there is only one Farm.

Optional field values to set:

Name	Type	Description
NewName	string	Name of the Farm. Default='' Max Length=50
Description	string	User description. Default='' Max Length=250
AutoAddEnabled	bool	True when Auto Add is enabled, false otherwise. Default=false
AuditingEnabled	bool	True when Auditing is enabled, false otherwise. Default=false

Name	Type	Description
LastAuditArchiveDate	DateTime	Last date of Audit Trail data that was Archived. Has the date. Default=Empty
DefaultSiteId	Guid	GUID of the Site to place new Devices into automatically. Not used with defaultSiteName. Default=00000000-0000-0000-0000-000000000000
DefaultSiteName	string	Name of the Site to place new Devices into automatically. Not used with DefaultSiteId. Default=’”’
OfflineDatabaseSupportEnabled	bool	True when Offline Database Support is enabled, false otherwise. Default=false
LicenseServer	string	License server name. Default=’”’Max Length=255
LicenseServerPort	uint	License server port. Min=1025, Max=65534, Default=27000
LicenseTradeUp	bool	License server trade up, when set to true. Default=false
LicenseSKU	uint	LicenseSKU. 0 for on-premises, 1 for cloud. Min=0, Max=1, Default=0
LicenseWebServicesPort	uint	The license server web services for licensing port. Min=1025, Max=65534, Default=8083
AutomaticMergeEnabled	bool	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	uint	Maximum number a versions of a Disk that can exist before a merge will automatically occur. Min=3, Max=50, Default=5

Name	Type	Description
MergeMode	uint	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
FarmEpoch	UInt64	Timestamp when the Data Encryption Key (DEK) was created and stored in the Credential Wallet. Default=0
EncryptionStatus	uint	The current status for database encryption. 0 = Idle, 1 = Distributing Keys, 2= Re-encrypting database Min=0, Max=2, Default=0
DefaultDeviceLogLevel	uint	The default logging level used by newly created target devices. Values are: 0 (None), 1 (Fatal), 2 (Error), 3 (Warning), 4 (Info), and 5 (Debug). Min=0, Max=6, Default=4
IsUsageDataScrambled	bool	Indicates whether the sensitive information needs to be scrambled (true) or not (false) when uploading usage telemetry data. Default = true. Default=true
LicenseServerCertThumbprint	string	Stores the license server's server certificate thumbprint when the server certificate is explicitly accepted by the administrator. NULL if no certificate is explicitly accepted. Default=""Max Length=65535

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.

Examples

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

Name	Type	Description
Name or FarmViewName	string	name of the Farm View. Max Length=50
Description	string	User description. Default=”” Max Length=250

When FarmView is not passed, the parameters below are used:
One of these is required

Name	Type	Description
Guid or FarmViewId	Guid	GUID of the Farm View to Set.
Name or FarmViewName	string	Name of the Farm View to Set.

Optional field values to set:

Name	Type	Description
NewName	string	name of the Farm View. Max Length=50
Description	string	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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator File to Override the Production Version for.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

Optional

Name	Type	Description
Version	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

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.

Examples

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

Name	Type	Description
Name or ServerName	string	Computer name with no spaces. ASCII computer name characters Max Length=21
Description	string	User description. Default=’”’ Max Length=250
AdMaxPasswordAge	uint	Number of days before a password expires. Min=1, Max=30, Default=7
LicenseTimeout	uint	Amount of seconds before a license times out. Min=15, Max=300, Default=30

Name	Type	Description
VDiskCreatePacing	uint	VDisk create time pacing in milliseconds. Min=0, Max=5, Default=0
FirstPort	uint	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	uint	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	uint	Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8
BuffersPerThread	uint	Number of buffers per worker thread. Min=1, Max=128, Default=24
ServerCacheTimeout	uint	Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8
IoBurstSize	uint	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	uint	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

Name	Type	Description
MaxBootDevicesAllowed	uint	Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500
MaxBootSeconds	uint	Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60
BootPauseSeconds	uint	Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10
AdMaxPasswordAgeEnabled	bool	Age the password, when set to true. Default=false
EventLoggingEnabled	bool	Enable event logging, when set to true. Default=true
NonBlockingIoEnabled	bool	Use non-Blocking IO, when set to true. Default=true
Ip	string[]	One or more streaming IPv4 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv4 addresses
Ipv6	string[]	One or more streaming IPv6 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv6 addresses
InitialQueryConnectionPoolSize	uint	Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50
InitialTransactionConnectionPoolSize	uint	Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50

Name	Type	Description
MaxQueryConnectionPoolSize	uint	Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000
MaxTransactionConnectionPoolSize	uint	Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000
RefreshInterval	uint	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	uint	Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300
BusyDbConnectionRetryCount	uint	Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2
BusyDbConnectionRetryInterval	uint	Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25
LocalConcurrentIoLimit	uint	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	uint	Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4

Name	Type	Description
LogLevel	uint	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	uint	Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5
LogFileBackupCopiesMax	uint	Maximum number of log file backups. Min=1, Max=50, Default=4
PowerRating	float	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
LastCeipUploadAttempt	DateTime	Time that this server last attempted a CEIP upload. Default=Empty
LastBugReportAttempt	DateTime	Time that this server last attempted to upload or generate a bug report bundle. Default=Empty
LastBugReportStatus	string	Status of the last bug report on this server. Default='"'Max Length=250
LastBugReportResult	string	Status of the last bug report on this server. Default='"'Max Length=4000
LastBugReportSummary	string	Summary of the last bug report on this server. Default='"'Max Length=250

Name	Type	Description
ServerEpoch	UInt64	Timestamp when the registry data for that server has been re-encrypted with updated encryption scheme. Default=0
SecurityPolicy	uint	SecurityPolicy Default=0
QuicSettings	string	QuicSettings Default=’”’Max Length=65535
QuicPort	uint	QuicPort Default=0
QuicServerCertThumbPrint	string	QuicServerCertThumbPrint Default=’”’Max Length=65535
QuicCipherSuites	uint	QuicCipherSuites Default=’”’
QuicStreamPayloadLength	uint	QuicStreamPayloadLength Default=0
TpmKeyType	uint	TPM key type, TPM_ALG_RSA(1), TPM_ALG_ECC(0x23) Default=0
TpmHashAlg	uint	TPM hash algorithm, TPM_ALG_SHA256(0xb), TPM_ALG_SHA384(0xc), TPM_ALG_SHA512(0xd) Default=0
TpmCurveId	uint	TPM ECC curve ID TPM_ECC_NIST_P256(3), TPM_ECC_NIST_P384(4), TPM_ECC_NIST_P521(5) Default=0

When Server is not passed, the parameters below are used:
One of these is required

Name	Type	Description
Guid or ServerId	Guid	GUID of the Server to Set.
Name or ServerName	string	Name of the Server to Set.

Optional field values to set:

Name	Type	Description
NewName	string	Computer name with no spaces. ASCII computer name characters Max Length=21
Description	string	User description. Default=’”’ Max Length=250
AdMaxPasswordAge	uint	Number of days before a password expires. Min=1, Max=30, Default=7
LicenseTimeout	uint	Amount of seconds before a license times out. Min=15, Max=300, Default=30
VDiskCreatePacing	uint	VDisk create time pacing in milliseconds. Min=0, Max=5, Default=0
FirstPort	uint	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	uint	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	uint	Number of worker threads per IO port. Required that (threadPerPort * numberPorts * numberIPs) <= 1000. Min=1, Max=60, Default=8
BuffersPerThread	uint	Number of buffers per worker thread. Min=1, Max=128, Default=24
ServerCacheTimeout	uint	Number of seconds to wait before considering another Server is down. Min=5, Max=60, Default=8

Name	Type	Description
IoBurstSize	uint	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	uint	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	uint	Maximum number of Devices allowed to boot simultaneously. Min=1, Max=1000, Default=500
MaxBootSeconds	uint	Maximum number of seconds for a Device to boot. Min=10, Max=900, Default=60
BootPauseSeconds	uint	Number of seconds that a Device will pause during login if its server busy. Min=1, Max=60, Default=10
AdMaxPasswordAgeEnabled	bool	Age the password, when set to true. Default=false
EventLoggingEnabled	bool	Enable event logging, when set to true. Default=true
NonBlockingIoEnabled	bool	Use non-Blocking IO, when set to true. Default=true
Ip	string[]	One or more streaming IPv4 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv4 addresses

Name	Type	Description
Ipv6	string[]	One or more streaming IPv6 Addresses. If more than one address is included they are comma delimited. NOTE that an empty string results in no configured IPv6 addresses
InitialQueryConnectionPoolSize	uint	Initial size of database connection pool for non-transactional queries. Min=1, Max=1000, Default=50
InitialTransactionConnectionPoolSize	uint	Initial size of database connection pool for transactional queries. Min=1, Max=1000, Default=50
MaxQueryConnectionPoolSize	uint	Maximum size of database connection pool for non-transactional queries. Min=1, Max=32767, Default=1000
MaxTransactionConnectionPoolSize	uint	Maximum size of database connection pool for transactional queries. Min=1, Max=32767, Default=1000
RefreshInterval	uint	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	uint	Interval, in number of seconds, a connection should go unused before it is to be released. Min=0, Max=32767, Default=300
BusyDbConnectionRetryCount	uint	Number of times a failed database connection will be retried. Min=0, Max=32767, Default=2

Name	Type	Description
BusyDbConnectionRetryInterval	uint	Interval, in number of milliseconds, the server should wait before retrying to connect to a database. Min=0, Max=10000, Default=25
LocalConcurrentIoLimit	uint	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	uint	Maximum concurrent IO transactions it performs for vDisks that are remote. A value of 0 disables the feature. Min=0, Max=128, Default=4
LogLevel	uint	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	uint	Maximum size log files can reach in Megabytes. Min=1, Max=50, Default=5
LogFileBackupCopiesMax	uint	Maximum number of log file backups. Min=1, Max=50, Default=4
PowerRating	float	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
LastCeipUploadAttempt	DateTime	Time that this server last attempted a CEIP upload. Default=Empty

Name	Type	Description
LastBugReportAttempt	DateTime	Time that this server last attempted to upload or generate a bug report bundle. Default=Empty
LastBugReportStatus	string	Status of the last bug report on this server. Default=’”’Max Length=250
LastBugReportResult	string	Status of the last bug report on this server. Default=’”’Max Length=4000
LastBugReportSummary	string	Summary of the last bug report on this server. Default=’”’Max Length=250
ServerEpoch	UInt64	Timestamp when the registry data for that server has been re-encrypted with updated encryption scheme. Default=0
SecurityPolicy	uint	SecurityPolicy Default=0
QuicSettings	string	QuicSettings Default=’”’Max Length=65535
QuicPort	uint	QuicPort Default=0
QuicServerCertThumbPrint	string	QuicServerCertThumbPrint Default=’”’Max Length=65535
QuicCipherSuites	uint	QuicCipherSuites Default=’”’
QuicStreamPayloadLength	uint	QuicStreamPayloadLength Default=0
TpmKeyType	uint	TPM key type, TPM_ALG_RSA(1), TPM_ALG_ECC(0x23) Default=0
TpmHashAlg	uint	TPM hash algorithm, TPM_ALG_SHA256(0xb), TPM_ALG_SHA384(0xc), TPM_ALG_SHA512(0xd) Default=0

Name	Type	Description
TpmCurveId	uint	TPM ECC curve ID TPM_ECC_NIST_P256(3), TPM_ECC_NIST_P384(4), TPM_ECC_NIST_P521(5) Default=0

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.

Examples

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

Name	Type	Description
Enabled	bool	Automatically update the BIOS on the target device with these setting when set to true, otherwise do not use these settings. Default=false
DhcpEnabled	bool	Use DHCP to retrieve target device IP when set to true, otherwise use the static domain, dnsIpAddresstrue and dnsIpAddress2 settings. Default=true
Lookup	bool	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	bool	Display verbose diagnostic information when set to true. Default=false

Name	Type	Description
InterruptSafeMode	bool	Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false
PaeMode	bool	PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false
BootFromHdOnFail	bool	For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false
RecoveryTime	uint	When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50
PollingTimeout	uint	Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000
GeneralTimeout	uint	Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000
Name or ServerName	string	Host to use for DNS lookup. Only used when Lookup is true. Default=IMAGESERVER1
Bootserver1_Ip	string	1st boot server IP. Only used when Lookup is false.
Bootserver1_Port	uint	1st boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Bootserver2_Ip	string	2nd boot server IP. Only used when Lookup is false. Default=0.0.0.0
Bootserver2_Port	uint	2nd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910

Name	Type	Description
Bootserver3_Ip	string	3rd boot server IP. Only used when Lookup is false. Default=0.0.0.0
Bootserver3_Port	uint	3rd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Bootserver4_Ip	string	4th boot server IP. Only used when Lookup is false. Default=0.0.0.0
Bootserver4_Port	uint	4th boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Domain	string	Domain of the primary and secondary DNS servers. Only used when DhcpEnabled is false.
DnsIpAddress1	string	Primary DNS server IP. Only used when DhcpEnabled is false.
DnsIpAddress2	string	Secondary DNS server IP. Only used when DhcpEnabled is false.

When ServerBiosBootstrap is not passed, the parameters below are used:
One of these is required

Name	Type	Description
Guid or ServerId	Guid	GUID of the Server to Set the dell_bios.bin BIOS bootstrap file.
Name or ServerName	string	Name of the Server to Set the dell_bios.bin BIOS bootstrap file.

Optional field values to set:

Name	Type	Description
Enabled	bool	Automatically update the BIOS on the target device with these setting when set to true, otherwise do not use these settings. Default=false
DhcpEnabled	bool	Use DHCP to retrieve target device IP when set to true, otherwise use the static domain, dnsIpAddresstrue and dnsIpAddress2 settings. Default=true
Lookup	bool	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	bool	Display verbose diagnostic information when set to true. Default=false
InterruptSafeMode	bool	Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false
PaeMode	bool	PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false
BootFromHdOnFail	bool	For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false

Name	Type	Description
RecoveryTime	uint	When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50
PollingTimeout	uint	Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000
GeneralTimeout	uint	Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000
NewName	string	Host to use for DNS lookup. Only used when Lookup is true. Default=IMAGESERVER1
Bootserver1_Ip	string	1st boot server IP. Only used when Lookup is false.
Bootserver1_Port	uint	1st boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Bootserver2_Ip	string	2nd boot server IP. Only used when Lookup is false. Default=0.0.0.0
Bootserver2_Port	uint	2nd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Bootserver3_Ip	string	3rd boot server IP. Only used when Lookup is false. Default=0.0.0.0
Bootserver3_Port	uint	3rd boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Bootserver4_Ip	string	4th boot server IP. Only used when Lookup is false. Default=0.0.0.0

Name	Type	Description
Bootserver4_Port	uint	4th boot server port. Only used when Lookup is false. Min=1025, Max=65536, Default=6910
Domain	string	Domain of the primary and secondary DNS servers. Only used when DhcpEnabled is false.
DnsIpAddress1	string	Primary DNS server IP. Only used when DhcpEnabled is false.
DnsIpAddress2	string	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.

Examples

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

Name	Type	Description
Bootserver1_Ip	string	1st boot server IP.
Bootserver1_Netmask	string	1st boot server netmask. Default=0.0.0.0
Bootserver1_Gateway	string	1st boot server gateway. Default=0.0.0.0
Bootserver1_Port	uint	1st boot server port. Min=1025, Max=65536, Default=6910
Bootserver2_Ip	string	2nd boot server IP. Default=0.0.0.0
Bootserver2_Netmask	string	2nd boot server netmask. Default=0.0.0.0
Bootserver2_Gateway	string	2nd boot server gateway. Default=0.0.0.0
Bootserver2_Port	uint	2nd boot server port. Min=1025, Max=65536, Default=6910
Bootserver3_Ip	string	3rd boot server IP. Default=0.0.0.0
Bootserver3_Netmask	string	3rd boot server netmask. Default=0.0.0.0

Name	Type	Description
Bootserver3_Gateway	string	3rd boot server gateway. Default=0.0.0.0
Bootserver3_Port	uint	3rd boot server port. Min=1025, Max=65536, Default=6910
Bootserver4_Ip	string	4th boot server IP. Default=0.0.0.0
Bootserver4_Netmask	string	4th boot server netmask. Default=0.0.0.0
Bootserver4_Gateway	string	4th boot server gateway. Default=0.0.0.0
Bootserver4_Port	uint	4th boot server port. Min=1025, Max=65536, Default=6910
VerboseMode	bool	Display verbose diagnostic information when set to true. Default=false
InterruptSafeMode	bool	Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false
PaeMode	bool	PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false
BootFromHdOnFail	bool	For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false
RecoveryTime	uint	When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50
PollingTimeout	uint	Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000
GeneralTimeout	uint	Login general timeout in milliseconds. Min=1000, Max=60000, Default=5000

When ServerBootstrap is not passed, the parameters below are used:
One of these is required

Name	Type	Description
Guid or ServerId	Guid	GUID of the Server to Set the named bootstrap file on.
ServerName	string	Name of the Server to Set the named bootstrap file on.

This is required

Name	Type	Description
Name	string	

Optional field values to set:

Name	Type	Description
Bootserver1_Ip	string	1st boot server IP.
Bootserver1_Netmask	string	1st boot server netmask. Default=0.0.0.0
Bootserver1_Gateway	string	1st boot server gateway. Default=0.0.0.0
Bootserver1_Port	uint	1st boot server port. Min=1025, Max=65536, Default=6910
Bootserver2_Ip	string	2nd boot server IP. Default=0.0.0.0
Bootserver2_Netmask	string	2nd boot server netmask. Default=0.0.0.0
Bootserver2_Gateway	string	2nd boot server gateway. Default=0.0.0.0
Bootserver2_Port	uint	2nd boot server port. Min=1025, Max=65536, Default=6910
Bootserver3_Ip	string	3rd boot server IP. Default=0.0.0.0
Bootserver3_Netmask	string	3rd boot server netmask. Default=0.0.0.0

Name	Type	Description
Bootserver3_Gateway	string	3rd boot server gateway. Default=0.0.0.0
Bootserver3_Port	uint	3rd boot server port. Min=1025, Max=65536, Default=6910
Bootserver4_Ip	string	4th boot server IP. Default=0.0.0.0
Bootserver4_Netmask	string	4th boot server netmask. Default=0.0.0.0
Bootserver4_Gateway	string	4th boot server gateway. Default=0.0.0.0
Bootserver4_Port	uint	4th boot server port. Min=1025, Max=65536, Default=6910
VerboseMode	bool	Display verbose diagnostic information when set to true. Default=false
InterruptSafeMode	bool	Interrupt safe mode (use if target device hangs during boot) when set to true. Default=false
PaeMode	bool	PAE mode (use if PAE enabled in boot.ini of target device) when set to true. Default=false
BootFromHdOnFail	bool	For network recovery reboot to hard drive when set to true, restore network connection when set to false. Default=false
RecoveryTime	uint	When bootFromHdOnFail is 1, this is the number of seconds to wait before reboot to hard drive. Min=10, Max=60000, Default=50
PollingTimeout	uint	Login polling timeout in milliseconds. Min=1000, Max=60000, Default=5000
GeneralTimeout	uint	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.

Examples

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

Name	Type	Description
Path	string	Directory path that the Server uses to access the Store. Default=’”’Max Length=255
CachePath	string[]	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 is required

Name	Type	Description
Guid or ServerId	Guid	GUID of a Server that uses the path to get to the Store.
Name or ServerName	string	Name of a Server that uses the path to get to the Store.

One of these is required

Name	Type	Description
StoreId	Guid	
StoreName	string	

Optional field values to set:

Name	Type	Description
Path	string	Directory path that the Server uses to access the Store. Default=’”’Max Length=255

Name	Type	Description
CachePath	string[]	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.

Examples

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

Name	Type	Description
Name or SiteName	string	Name of the Site. Max Length=50
Description	string	User description. Default=’”’ Max Length=250
DefaultCollectionId	Guid	GUID of the Collection to place new Devices into automatically. Not used with defaultCollectionName. Default=00000000-0000-0000-0000-000000000000
DefaultCollectionName	string	Name of the Collection to place new Devices into automatically. Not used with DefaultCollectionId. Default=’”’
InventoryFilePollingInterval	uint	The number of seconds between polls for Disk changes in the Stores. Min=1, Max=600, Default=60
EnableDiskUpdate	bool	True when Disk Updated is enabled for the Site, false otherwise. Default=false
DiskUpdateServerId	Guid	GUID of the Disk Update Server for the Site. Not used with DiskUpdateServerName. Default=00000000-0000-0000-0000-000000000000
DiskUpdateServerName	string	Name of the Disk Update Server for the Site. Not used with DiskUpdateServerId. Default=’”’

Name	Type	Description
MakUser	string	User name used for MAK activation. Default=’”’Max Length=64
MakPassword	string	User password used for MAK activation. Default=’”’Max Length=64
EnableXsProxy	string	Enable XenServerProxy when set to 1 Default=’”’
VirtualHostingPoolId	Guid	GUID of the VirtualHostingPool object.
VirtualHostingPoolName	string	Name of the VirtualHostingPool object.
XsPvsSiteUuid	string	GUID of the XenServer PVS Site.

When Site is not passed, the parameters below are used:
One of these is required

Name	Type	Description
Guid or SiteId	Guid	GUID of the Site to Set.
Name or SiteName	string	Name of the Site to Set.

Optional field values to set:

Name	Type	Description
NewName	string	Name of the Site. Max Length=50
Description	string	User description. Default=’”’ Max Length=250
DefaultCollectionId	Guid	GUID of the Collection to place new Devices into automatically. Not used with defaultCollectionName. Default=00000000-0000-0000-0000-000000000000

Name	Type	Description
DefaultCollectionName	string	Name of the Collection to place new Devices into automatically. Not used with DefaultCollectionId. Default=’”’
InventoryFilePollingInterval	uint	The number of seconds between polls for Disk changes in the Stores. Min=1, Max=600, Default=60
EnableDiskUpdate	bool	True when Disk Updated is enabled for the Site, false otherwise. Default=false
DiskUpdateServerId	Guid	GUID of the Disk Update Server for the Site. Not used with DiskUpdateServerName. Default=00000000-0000-0000-0000-000000000000
DiskUpdateServerName	string	Name of the Disk Update Server for the Site. Not used with DiskUpdateServerId. Default=’”’
MakUser	string	User name used for MAK activation. Default=’”’Max Length=64
MakPassword	string	User password used for MAK activation. Default=’”’Max Length=64
EnableXsProxy	string	Enable XenServerProxy when set to 1 Default=’”’
VirtualHostingPoolId	Guid	GUID of the VirtualHostingPool object.
VirtualHostingPoolName	string	Name of the VirtualHostingPool object.
XsPvsSiteUuid	string	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.

Examples

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

Name	Type	Description
Name or SiteViewName	string	Name of the Site View. Max Length=50

Name	Type	Description
Description	string	User description. Default=’”’ Max Length=250

When SiteView is not passed, the parameters below are used:
This is required

Name	Type	Description
Guid or SiteViewId	Guid	GUID of the Site View to Set.

or this is required & resolution

Name	Type	Description
Name or SiteViewName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

Optional field values to set:

Name	Type	Description
NewName	string	Name of the Site View. Max Length=50
Description	string	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.

Examples

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

Name	Type	Description
Name or StoreName	string	Name of the Store. Max Length=50

Name	Type	Description
SiteId	Guid	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	string	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	string	User description. Default=’”’ Max Length=250
Path	string	Default directory path that the Servers use to access this Store. Max Length=255
CachePath	string[]	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 is required

Name	Type	Description
Guid or StoreId	Guid	GUID of the Store to Set.
Name or StoreName	string	Name of the Store to Set.

Optional field values to set:

Name	Type	Description
NewName	string	Name of the Store. Max Length=50
SiteId	Guid	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	string	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	string	User description. Default=”” Max Length=250
Path	string	Default directory path that the Servers use to access this Store. Max Length=255
CachePath	string[]	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.

Examples

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

Name	Type	Description
Name or UpdateTaskName	string	Name of the Update Task. It is unique within the Site. Max Length=50
Description	string	User description. Default=”” Max Length=250
Enabled	bool	True when it will be processed, false otherwise. Default=true
Hour	uint	The hour of the day to perform the task. Min=0, Max=23, Default=0

Name	Type	Description
Minute	uint	The minute of the hour to perform the task. Min=0, Max=59, Default=0
Recurrence	uint	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	uint	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	uint[]	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	uint	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	string	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	string	Script file to run before the update starts. Default=’”’Max Length=255

Name	Type	Description
PreVmScript	string	Script file to run before the VM is loaded. Default=’”’Max Length=255
PostUpdateScript	string	Script file to run after the update finishes. Default=’”’Max Length=255
PostVmScript	string	Script file to run after the VM is unloaded. Default=’”’Max Length=255
Domain	string	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	string	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

Name	Type	Description
PostUpdateApprove	uint	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 is required

Name	Type	Description
Guid or UpdateTaskId	Guid	GUID of the Update Task to Set.

or this is required & resolution

Name	Type	Description
Name or UpdateTaskName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

Optional field values to set:

Name	Type	Description
NewName	string	Name of the Update Task. It is unique within the Site. Max Length=50
Description	string	User description. Default=’”’ Max Length=250

Name	Type	Description
Enabled	bool	True when it will be processed, false otherwise. Default=true
Hour	uint	The hour of the day to perform the task. Min=0, Max=23, Default=0
Minute	uint	The minute of the hour to perform the task. Min=0, Max=59, Default=0
Recurrence	uint	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	uint	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	uint[]	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	uint	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	string	Esd to use. Valid values are SCCM or WSUS. If no value, a custom script is run on the client. Default=’”’Max Length=50

Name	Type	Description
PreUpdateScript	string	Script file to run before the update starts. Default=’”’Max Length=255
PreVmScript	string	Script file to run before the VM is loaded. Default=’”’Max Length=255
PostUpdateScript	string	Script file to run after the update finishes. Default=’”’Max Length=255
PostVmScript	string	Script file to run after the VM is unloaded. Default=’”’Max Length=255
Domain	string	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

Name	Type	Description
OrganizationUnit	string	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	uint	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.

Examples

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

Name	Type	Description
Name or VirtualHostingPoolName	string	Name of the Virtual Hosting Pool. It is unique within the Site. Max Length=50

Name	Type	Description
Type	uint	Type of the Virtual Hosting Pool. 0 = Citrix XenServer, 1 = Microsoft SCVMM/Hyper-V, 2 = VMWare vSphere/ESX, 3 = Nutanix, 4 = Azure, 5 = GCP. Min=0, Max=255, Default=0
Description	string	User description. Default=’”’ Max Length=250
Server	string	Name or IP of the Host Server. Max Length=255
Port	uint	Port of the Host Server. Min=80, Max=65534, Default=80
Datacenter	string	Datacenter of the Virtual Hosting Pool. Default=’”’Max Length=250
UpdateLimit	uint	Number of updates at the same time. Min=2, Max=1000, Default=1000
UpdateTimeout	uint	Timeout for updates. Min=2, Max=240, Default=60
ShutdownTimeout	uint	Timeout for shutdown. Min=2, Max=30, Default=10
UserName	string	Name to use when logging into the Server.
Password	string	Password to use when logging into the Server.
XdHostingUnitUuid	Guid	UUID of XenDesktop Hosting Unit Default=00000000-0000-0000-0000-000000000000
PrepopulateEnabled	bool	Enable prepopulate when set to true Default=false
XsPvsSiteUuid	Guid	UUID of XenServer PVS_site Default=00000000-0000-0000-0000-000000000000
PlatformVersion	string	Hypervisor Host Version Default=’”’Max Length=250

Name	Type	Description
XdHcHypervisorConnectionName	string	Hypervisor Connection Name for HCL Connection Details object Default=’”’Max Length=250
XdHcHypervisorConnectionUid	string	Hypervisor Connection Uid for HCL Connection Details object Default=’”’Max Length=250
XdHcRevision	string	Revision for HCL Connection Details object Default=’”’Max Length=250
XdHcCustomProperties	string	Custom Properties for HCL Connection Details object Default=’”’Max Length=65535
XdHcSslThumbprints	string	Ssl Thumbprints for HCL Connection Details object Default=’”’Max Length=65535
CredentialsUid	string	UUID of any record associated with the credentials Default=’”’ Max Length=65535
DdcType	uint	Type of the DDC. 0 = Unset, 1 = Citrix Cloud, 2 = customer-managed. Min=0, Max=2, Default=0
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0
DisableHostXsProxy	string	True to disable PVS-Accelerator Default=’”’

When VirtualHostingPool is not passed, the parameters below are used:
This is required

Name	Type	Description
Guid or VirtualHostingPoolId	Guid	GUID of the Virtual Hosting Pool to Set.

or this is required & resolution

Name	Type	Description
Name or VirtualHostingPoolName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

Optional field values to set:

Name	Type	Description
NewName	string	Name of the Virtual Hosting Pool. It is unique within the Site. Max Length=50
Type	uint	Type of the Virtual Hosting Pool. 0 = Citrix XenServer, 1 = Microsoft SCVMM/Hyper-V, 2 = VMWare vSphere/ESX, 3 = Nutanix, 4 = Azure, 5 = GCP. Min=0, Max=255, Default=0
Description	string	User description. Default=’”’ Max Length=250
Server	string	Name or IP of the Host Server. Max Length=255
Port	uint	Port of the Host Server. Min=80, Max=65534, Default=80
Datacenter	string	Datacenter of the Virtual Hosting Pool. Default=’”’Max Length=250
UpdateLimit	uint	Number of updates at the same time. Min=2, Max=1000, Default=1000

Name	Type	Description
UpdateTimeout	uint	Timeout for updates. Min=2, Max=240, Default=60
ShutdownTimeout	uint	Timeout for shutdown. Min=2, Max=30, Default=10
UserName	string	Name to use when logging into the Server.
Password	string	Password to use when logging into the Server.
XdHostingUnitUuid	Guid	UUID of XenDesktop Hosting Unit Default=00000000-0000-0000-0000-000000000000
PrepopulateEnabled	bool	Enable prepopulate when set to true Default=false
XsPvsSiteUuid	Guid	UUID of XenServer PVS_site Default=00000000-0000-0000-0000-000000000000
PlatformVersion	string	Hypervisor Host Version Default=’”’Max Length=250
XdHcHypervisorConnectionName	string	Hypervisor Connection Name for HCL Connection Details object Default=’”’Max Length=250
XdHcHypervisorConnectionUid	string	Hypervisor Connection Uid for HCL Connection Details object Default=’”’Max Length=250
XdHcRevision	string	Revision for HCL Connection Details object Default=’”’Max Length=250
XdHcCustomProperties	string	Custom Properties for HCL Connection Details object Default=’”’Max Length=65535
XdHcSslThumbprints	string	Ssl Thumbprints for HCL Connection Details object Default=’”’Max Length=65535
CredentialsUid	string	UUID of any record associated with the credentials Default=’”’ Max Length=65535

Name	Type	Description
DdcType	uint	Type of the DDC. 0 = Unset, 1 = Citrix Cloud, 2 = customer-managed. Min=0, Max=2, Default=0
EncryptionEpoch	UInt64	Timestamp of the encryption Key used to encrypt fields Default=0
DisableHostXsProxy	string	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.

Examples

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 PvsXD-Site.

Required
PvsXDSite XDSite: PvsXDSite object(s) with changed property value(s) to be set. The object(s) can come from a pipeline.
These values are in the PvsXDSite object, and only will be set if the value has changed.

Name	Type	Description
ConfigServices	string[]	XenDesktop Server addresses. Max Length=2000

When XDSite is not passed, the parameters below are used:
This is required

Name	Type	Description
Guid or XdSiteId	Guid	GUID of the XenDesktop Site to Set.

Optional field values to set:

Name	Type	Description
ConfigServices	string[]	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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to apply Auto Update.
Name or ServerName	string[]	Name of the Server to apply Auto Update.
SiteId	Guid[]	GUID of the Site to apply Auto Update on all Servers.
SiteName	string[]	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 pipeline:
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.

Examples

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-PvsCreateBDM

Create a BDM disk file.

This is required

Name	Type	Description
DiskFileName	string[]	BDM filename

This is required

Name	Type	Description
VhdFileSize	uint	

One of these is required

Name	Type	Description
ServerIpList	string[]	
ServerFqdn	string[]	

Optional

Name	Type	Description
ServerPort	string[]	
BootNicIndex	string[]	
UsingIpv6	string[]	
DnsIpAddress1	string[]	
DnsIpAddress2	string[]	

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.

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 is required

Name	Type	Description
Name	string	Name of the Disk file to be created.

This is required

Name	Type	Description
Size	UInt64	

One of these is required

Name	Type	Description
StoreId	Guid	
StoreName	string	

One of these is required

Name	Type	Description
SiteId	Guid	
SiteName	string	

These are optional

Name	Type	Description
ServerId	Guid	
ServerName	string	

Optional

Name	Type	Description
Description	string	
Enabled	SwitchParameter	
Format	SwitchParameter	
Type	SwitchParameter	
VhdBlockSize	uint	
LogicalSectorSize	uint	

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 pipeline:
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.

Examples

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
    {

```



```
15          %percentFinished.ToString() + "% finished"                  #
                display percent finished
16          Start-Sleep -seconds 10                                     #
                wait 10 seconds more
17      }
18
19  }
20
21  "Successful"
```

EXAMPLE 2: Start-PvsCreateDisk for VHD to PvsDiskLocator TThis example creates a Fixed VHD Disk that has a size of 20 GB.

```
1   $thePvsDiskLocator = Start-PvsCreateDisk -Name theDiskName -Size 20480
        -StoreName theStore -SiteName theSite
2   while ($thePvsDiskLocator -eq $null)                                # while
        the create is processing
3   {
4
5       %percentFinished = Get-PvsCreateDiskStatus -Name theDiskName -
            StoreName theStore  # get percent finished or DiskLocator when
            done
6       if (%percentFinished.GetType().Name == "PvsDiskLocator")
7       {
8
9           $thePvsDiskLocator = %percentFinished
10      }
11
12      else
13      {
14
15          %percentFinished.ToString() + "% finished"                  #
                display percent finished
16          Start-Sleep -seconds 10                                     #
                wait 10 seconds more
17      }
18
19  }
20
21  "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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Boot.
Name or DeviceName	string[]	Name of the Device to Boot.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Boot.
CollectionId	Guid	GUID of the Collection to Boot all Devices.
SiteViewId	Guid	GUID of the Site View to Boot all Devices.
FarmViewId	Guid	GUID of the Farm View to Boot all Devices.
FarmViewName	string	Name of the Farm View to Boot all Devices.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to start to use a temporary disk version.
Name or DeviceName	string[]	Name of the Device to start to use a temporary disk version.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to start to use a temporary disk version.

This is optional

Name	Type	Description
DiskLocatorId	Guid[]	

or this is optional & resolution

Name	Type	Description
DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Reboot.
Name or DeviceName	string[]	Name of the Device to Reboot.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Reboot.
CollectionId	Guid	GUID of the Collection to Reboot all Devices.
SiteViewId	Guid	GUID of the Site View to Reboot all Devices.
FarmViewId	Guid	GUID of the Farm View to Reboot all Devices.
FarmViewName	string	Name of the Farm View to Reboot all Devices.
DiskLocatorId	Guid	GUID of the DiskLocator to Reboot all Devices.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	
DiskLocatorName	string	

Optional

Name	Type	Description
Message	string	
Delay	uint	
Version	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:

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.

Examples

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-PvsDeviceReconnect

Reconnect 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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Reconnect.
Name or DeviceName	string[]	Name of the Device to Reconnect.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Reconnect.
CollectionId	Guid	GUID of the Collection to Reconnect all Devices.
SiteViewId	Guid	GUID of the Site View to Reconnect all Devices.
FarmViewId	Guid	GUID of the Farm View to Reconnect all Devices.
FarmViewName	string	Name of the Farm View to Reconnect all Devices.
DiskLocatorId	Guid	GUID of the DiskLocator to Reconnect all Devices.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	
DiskLocatorName	string	

Optional

Name	Type	Description
Version	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:
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.

Examples

EXAMPLE 1: Start-PvsDeviceReconnect for DeviceName to PvsTask

```
$thePvsTask = Start-PvsDeviceReconnect -DeviceName theDevice # start
    the task
while ($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-PvsDeviceReconnect for DeviceMac

```
Start-PvsDeviceReconnect -DeviceMac "00-11-22-33-44-55"
```

EXAMPLE 3: Start-PvsDeviceReconnect for FarmViewName

```
Start-PvsDeviceReconnect -FarmViewName theFarmView
```

EXAMPLE 4: Start-PvsDeviceReconnect for CollectionName

```
Start-PvsDeviceReconnect -CollectionName theCollection -SiteName
   theSite
```

CollectionId can be used instead of CollectionName so that the SiteName or SiteId is not also needed.

EXAMPLE 5: Start-PvsDeviceReconnect for SiteViewName

```
Start-PvsDeviceReconnect -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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Shutdown.
Name or DeviceName	string[]	Name of the Device to Shutdown.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to Shutdown.
CollectionId	Guid	GUID of the Collection to Shutdown all Devices.
SiteViewId	Guid	GUID of the Site View to Shutdown all Devices.
FarmViewId	Guid	GUID of the Farm View to Shutdown all Devices.
FarmViewName	string	Name of the Farm View to Shutdown all Devices.
DiskLocatorId	Guid	GUID of the DiskLocator to Shutdown all Devices.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	
DiskLocatorName	string	

Optional

Name	Type	Description
Message	string	
Delay	uint	
Version	uint	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid	
StoreName	string	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to Update.
Name or DeviceName	string[]	Name of the Device to Update.
CollectionId	Guid	GUID of the Collection to Update all BDM Devices.

Name	Type	Description
SiteViewId	Guid	GUID of the Site View to Update all BDM Devices.
FarmViewId	Guid	GUID of the Farm View to Update all BDM Devices.
FarmViewName	string	Name of the Farm View to Update all BDM Devices.

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Message	string	Message to display on the Device(s).

One of these is required

Name	Type	Description
DeviceId	Guid[]	
DeviceName	string[]	
DeviceMac	PvsPhysicalAddress[]	
CollectionId	Guid	
SiteViewId	Guid	
FarmViewId	Guid	
FarmViewName	string	

or one of these is required & resolutions

Name	Type	Description
CollectionName	string	
SiteViewName	string	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid	
SiteName	string	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or SiteId	Guid	GUID of the Site.
ServerId	Guid	GUID of the Server.

or one of these is required & resolutions

Name	Type	Description
Name or SiteName	string	
ServerName	string	

This is required

Name	Type	Description
Summary	string	

This is optional

Name	Type	Description
SrNumber	string	

This is optional

Name	Type	Description
Description	string	

This is optional

Name	Type	Description
Path	string	

This is optional

Name	Type	Description
DateTime	DateTime	

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 pipeline:
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.

Examples

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-PvsRotateEncryptionKeys

Cycle the keys used to encrypt sensitive data

Examples

Cycle the keys used to encrypt sensitive data Cycle the keys used to encrypt sensitive data

```
Start-PvsRotateEncryptionKeys
```

Start-PvsStreamService

Start the Stream Service on a Server or all Servers in a Site.

One of these is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to start the Stream Service.

Name	Type	Description
Name or ServerName	string[]	Name of the Server to start the Stream Service.
SiteId	Guid[]	GUID of the Site to start the Stream Service on all Servers.
SiteName	string[]	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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or UpdateTaskId	Guid[]	GUID of the Update Task to Start.

or this is required & resolution

Name	Type	Description
Name or UpdateTaskName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Name	string[]	Name of the Disk file that is being created.

One of these is required

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or DeviceId	Guid[]	GUID of the Device to no longer use a temporary disk version for.
Name or DeviceName	string[]	Name of the Device to no longer use a temporary disk version for.
DeviceMac	PvsPhysicalAddress[]	MAC of the Device to no longer use a temporary disk version for.
DiskLocatorId	Guid[]	GUID of the DiskLocator to no longer use temporary disk versions for.
SiteId	Guid[]	GUID of the Site for resolution of the DiskLocatorName.
SiteName	string[]	Name of the Site for resolution of the DiskLocatorName.

or this is required & resolution

Name	Type	Description
DiskLocatorName	string[]	

This is optional & resolution

Name	Type	Description
Version	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	GUID of the Site for resolution of the DiskLocatorName.
SiteName	string[]	Name of the Site for resolution of the DiskLocatorName.

One of these resolutions when needed

Name	Type	Description
DiskLocatorId	Guid[]	GUID of the DiskLocator to no longer use temporary disk versions for.
DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to stop the Stream Service.
Name or ServerName	string[]	Name of the Server to stop the Stream Service.
SiteId	Guid[]	GUID of the Site to stop the Stream Service on all Servers.
SiteName	string[]	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 pipeline:
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.

Examples

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 is required

Name	Type	Description
TaskId	uint	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 pipeline:
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.

Examples

EXAMPLE 1: Stop-PvsTask for TaskId

```
Stop-PvsTask -TaskId 101
```

Stop-PvsUpdateTask

Cancel an active Update Task for an Update Device.

This is required

Name	Type	Description
Guid or UpdateTaskId	Guid[]	GUID of the Update Task to Cancel.

or this is required & resolution

Name	Type	Description
Name or UpdateTaskName	string[]	

One of these is required

Name	Type	Description
DeviceId	Guid[]	
DeviceName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

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 pipeline:
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.

Examples

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 is required

Name	Type	Description
Guid or ServerId	Guid[]	GUID of the Server to validate a Directory on.
Name or ServerName	string[]	Name of the Server to validate a Directory on.

This is required

Name	Type	Description
Path	string[]	

This is optional

Name	Type	Description
ReadOnly	SwitchParameter	

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 pipeline:
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.

Examples

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.

```
1  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.

Test-PvsVirtualHostConnection

Verify the virtual host connection

This is optional

Name	Type	Description
Password	string	Password to use when logging into the Virtual Host Server

or this is optional & resolution

Name	Type	Description
UserName	string	

Optional

Name	Type	Description
Guid or VirtualHostingPoolId	Guid	
Type	uint	
Server	string	
Port	uint	
Datacenter	string	

This resolution when needed

Name	Type	Description
Password	string	Password to use when logging into the Virtual Host Server

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.
Boolean: If successful, $true or $false is returned.

Examples

EXAMPLE 1: Test-PvsVirtualHostConnection for existing credentials

```
Test-PvsVirtualHostConnection -virtualHostingPoolId
   thevirtualHostingPoolId
```

EXAMPLE 2: Test-PvsVirtualHostConnection for specified credentials

```
Test-PvsVirtualHostConnection -virtualHostingPoolId
   thevirtualHostingPoolId -UserName theUser -Password thePassword
```

Unlock-PvsAllDisk

Remove all locks for a Disk.

This is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to remove all locks for the Disk.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:

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.

Examples

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 is required

Name	Type	Description
Guid or DiskLocatorId	Guid[]	GUID of the Disk Locator to remove a lock for the Disk.

or this is required & resolution

Name	Type	Description
Name or DiskLocatorName	string[]	

Optional

Name	Type	Description
OwnerId	Guid[]	

One of these resolutions when needed

Name	Type	Description
SiteId	Guid[]	
SiteName	string[]	

One of these resolutions when needed

Name	Type	Description
StoreId	Guid[]	
StoreName	string[]	

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 pipeline:
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.

Examples

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.

Examples

EXAMPLE 1: Update-PvsInventory

```
Update-PvsInventory
```

