MyWebUniversity

python PowerShell FPDF Library PDF generator

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'New-StoragePool'

PS C:\Users\wahid> Get-Help New-StoragePool

```
NAME
    New-StoragePool

SYNOPSIS
    Creates a new storage pool using a group of physical disks.


SYNTAX
    New-StoragePool [-AsJob] [-AutoWriteCacheSize <Boolean>] [-CimSession
    <CimSession[]>] [-EnclosureAwareDefault <Boolean>]
    [-FaultDomainAwarenessDefault {PhysicalDisk | StorageEnclosure |
    StorageScaleUnit | StorageChassis | StorageRack}] -FriendlyName <String>
    -InputObject <CimInstance[]> [-LogicalSectorSizeDefault <UInt64>]
    [-MediaTypeDefault {HDD | SSD | SCM}] [-OtherUsageDescription <String>]
    -PhysicalDisks <CimInstance[]> [-ProvisioningTypeDefault {Unknown | Thin |
    Fixed}] [-ResiliencySettingNameDefault <String>] [-ThrottleLimit <Int32>]
    [-Usage {Other | Unrestricted | ReservedForComputerSystem |
    ReservedAsDeltaReplicaContainer | ReservedForMigrationServices |
    ReservedForLocalReplicationServices | ReservedForRemoteReplicationServices |
    ReservedForSparing}] [-WriteCacheSizeDefault <UInt64>] [<CommonParameters>]
```

```
New-StoragePool [-StorageSubSystemFriendlyName] <String[]> [-AsJob]
[-AutoWriteCacheSize <Boolean>] [-CimSession <CimSession[]>]
[-EnclosureAwareDefault <Boolean>] [-FaultDomainAwarenessDefault {PhysicalDisk
| StorageEnclosure | StorageScaleUnit | StorageChassis | StorageRack}]
-FriendlyName <String> [-LogicalSectorSizeDefault <UInt64>] [-MediaTypeDefault
{HDD | SSD | SCM}] [-OtherUsageDescription <String>] -PhysicalDisks
<CimInstance[]> [-ProvisioningTypeDefault {Unknown | Thin | Fixed}]
[-ResiliencySettingNameDefault <String>] [-ThrottleLimit <Int32>] [-Usage
{Other | Unrestricted | ReservedForComputerSystem |
ReservedAsDeltaReplicaContainer | ReservedForMigrationServices |
ReservedForLocalReplicationServices | ReservedForRemoteReplicationServices |
ReservedForSparing}] [-WriteCacheSizeDefault <UInt64>] [<CommonParameters>]

New-StoragePool [-AsJob] [-AutoWriteCacheSize <Boolean>] [-CimSession
<CimSession[]>] [-EnclosureAwareDefault <Boolean>]
[-FaultDomainAwarenessDefault {PhysicalDisk | StorageEnclosure |
StorageScaleUnit | StorageChassis | StorageRack}] -FriendlyName <String>
[-LogicalSectorSizeDefault <UInt64>] [-MediaTypeDefault {HDD | SSD | SCM}]
[-OtherUsageDescription <String>] -PhysicalDisks <CimInstance[]>
[-ProvisioningTypeDefault {Unknown | Thin | Fixed}]
[-ResiliencySettingNameDefault <String>] -StorageSubSystemName <String[]>
[-ThrottleLimit <Int32>] [-Usage {Other | Unrestricted |
ReservedForComputerSystem | ReservedAsDeltaReplicaContainer |
ReservedForMigrationServices | ReservedForLocalReplicationServices |
ReservedForRemoteReplicationServices | ReservedForSparing}]
[-WriteCacheSizeDefault <UInt64>] [<CommonParameters>]

New-StoragePool [-AsJob] [-AutoWriteCacheSize <Boolean>] [-CimSession
<CimSession[]>] [-EnclosureAwareDefault <Boolean>]
[-FaultDomainAwarenessDefault {PhysicalDisk | StorageEnclosure |
StorageScaleUnit | StorageChassis | StorageRack}] -FriendlyName <String>
[-LogicalSectorSizeDefault <UInt64>] [-MediaTypeDefault {HDD | SSD | SCM}]
[-OtherUsageDescription <String>] -PhysicalDisks <CimInstance[]>
```

[-ProvisioningTypeDefault {Unknown | Thin | Fixed}]
[-ResiliencySettingNameDefault <String>] -StorageSubSystemUniqueId <String[]>
[-ThrottleLimit <Int32>] [-Usage {Other | Unrestricted |
ReservedForComputerSystem | ReservedAsDeltaReplicaContainer |
ReservedForMigrationServices | ReservedForLocalReplicationServices |
ReservedForRemoteReplicationServices | ReservedForSparing}]
[-WriteCacheSizeDefault <UInt64>] [<CommonParameters>]

DESCRIPTION

The New-StoragePool cmdlet creates a new storage pool using a group of physical disks exposed by a specific storage subsystem.

PARAMETERS

-AsJob [<SwitchParameter>]

Runs the cmdlet as a background job. Use this parameter to run commands that take a long time to complete.

-AutoWriteCacheSize <Boolean>

Indicates whether to set the size of the write-back cache to 1 GB for all types of storage spaces, which include simple, mirror, and parity, to create from the pool. If the number or size of the solid-state drives (SSDs) or journal disks in the storage pool is not sufficient and you specify a value of $True for this parameter, the cmdlet sets the write-back cache size to 0 for simple and mirror spaces, and to 32 MB for parity spaces.

-CimSession <CimSession[]>

Runs the cmdlet in a remote session or on a remote computer. Enter a computer name or a session object, such as the output of a New-CimSession (https://go.microsoft.com/fwlink/p/?LinkId=227967) or [Get-CimSession](https://go.microsoft.com/fwlink/p/?LinkId=227966)cmdlet.

The default is the current session on the local computer.

-EnclosureAwareDefault <Boolean>

Specifies the default allocation policy for virtual disks created in an enclosure-aware storage pool. For example, an enclosure-aware subsystem could balance each data copy of the virtual disk across multiple physical enclosures such that each enclosure contains a full data copy of the virtual disk.

-FaultDomainAwarenessDefault <FaultDomainType>

Specifies the default fault domain for new virtual disks created in this storage pool. The acceptable values for this parameter are:

- PhysicalDisk

- StorageScaleUnit

- StorageChassis

- StorageEnclosure

- StorageRack

The fault domain specifies at what level you want to be fault tolerant. For example, specify StorageScaleUnit to store data copies on separate nodes of a Storage Spaces Direct cluster. This cmdlet refers to nodes of a Storage Spaces Direct cluster as storage scale units because you can expand the scale of the cluster by adding more nodes.

-FriendlyName <String>

Specifies a friendly name for the storage pool to be created. The friendly name may be defined by a user and is not guaranteed to be unique.

-InputObject <CimInstance[]>

Specifies the input object that is used in a pipeline command.

-LogicalSectorSizeDefault <UInt64>

Specifies the default logical sector size to use for virtual disks created in this pool. Valid logical sector size values (in bytes) for virtual disks created by using the Windows Storage subsystem are 512 and 4096.

-MediaTypeDefault <MediaType>

-OtherUsageDescription <String>

Specifies the usage description for the storage pool.

-PhysicalDisks <CimInstance[]>

Accepts one or more PhysicalDisk objects as input. The Physical Disk CIM objects represent the physical disks to be added to the storage pool.

-ProvisioningTypeDefault <ProvisioningType>

Specifies the default type of provisioning for virtual disks created in this pool. The acceptable values for this parameter are: Unknown, Fixed or Thin.

-ResiliencySettingNameDefault <String>

Specifies the default resiliency setting (also known as storage layout) to use for virtual disks created in the specified storage pool. The supported resiliency settings vary by storage subsystem. For the Windows Storage subsystem, acceptable values are Mirror, Parity, and Simple. "Mirror" is the default value.

-StorageSubSystemFriendlyName <String[]>

Specifies the friendly name of the storage subsystem on which you want to create the storage pool.

-StorageSubSystemName <String[]>

Specifies the name of the storage subsystem (provided by the Storage Management) on which you want to create the storage pool.

-StorageSubSystemUniqueId <String[]>

Specifies the ID of the storage subsystem on which you want to create the storage pool

-ThrottleLimit <Int32>

Specifies the maximum number of concurrent operations that can be established to run the cmdlet. If this parameter is omitted or a value of `0` is entered, then Windows PowerShellr calculates an optimum throttle limit for the cmdlet based on the number of CIM cmdlets that are running on the computer. The throttle limit applies only to the current cmdlet, not to the session or to the computer.

-Usage <Usage>

Specifies the usage setting for the storage pool. The acceptable values for this parameter are: - Other

- ReservedAsDeltaReplicaContainer

- ReservedForComputerSystem

- ReservedForLocalReplicationServices

- ReservedForMigrationServices

- ReservedForRemoteReplicationServices

- ReservedForSparing

- Unrestricted

-WriteCacheSizeDefault <UInt64>

Specifies the default write-back cache size for virtual disks in the storage pool.

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

-- Example 1: Create a new storage pool using Storage Spaces --

```
PS C:\> $PhysicalDisks = (Get-PhysicalDisk -CanPool $True)
PS C:\> New-StoragePool -FriendlyName CompanyData
-StorageSubsystemFriendlyName "Windows Storage*" -PhysicalDisks $PhysicalDisks
```

The first line uses the Get-PhysicalDisk cmdlet to get all PhysicalDisk objects that are not yet in a (concrete) storage pool and assigns the array of objects to the `$PhysicalDisks` variable.

The second line creates a new storage pool using the `$PhysicalDisks` variable to specify the disks to include from the WindowsStorage subsystem (specified with a wildcard * to remove the need to modify the friendly name for different computers).

This example creates a new storage pool named CompanyData using the Storage Spaces subsytem, using the minimum parameters, and assuming that there are no other storage subsystems attached to the computer that have available disks.

Example 2: Create a new pool and set defaults for virtual disks

```
PS C:\> $PhysicalDisks = (Get-PhysicalDisk -CanPool $True)
PS C:\> New-StoragePool -FriendlyName CompanyData
-StorageSubsystemFriendlyName "Windows Storage*" -PhysicalDisks $PhysicalDisks
-ResiliencySettingNameDefault Mirror -ProvisioningTypeDefault Thin -Verbose
```

This example creates a new storage pool named CompanyData using the Windows Storage subsystem and sets default values for virtual disk creation.

Example 3: Create a new storage pool, virtual disk, partition, and volume

```
PS C:\> $PhysicalDisks = Get-StorageSubSystem -FriendlyName "Windows Storage*"
| Get-PhysicalDisk -CanPool $True
PS C:\> New-StoragePool -FriendlyName "CompanyData"
-StorageSubsystemFriendlyName "Windows Storage*" -PhysicalDisks $PhysicalDisks
| New-VirtualDisk -FriendlyName "UserData" -Size 100GB -ProvisioningType Thin
| Initialize-Disk -PassThru | New-Partition -AssignDriveLetter -UseMaximumSize
| Format-Volume
```

This example creates a new storage pool, and then makes use of the pipeline to create a new virtual disk in the pool, initialize the disk, create a new partition on the disk, and then format the new partition (volume).

Alternatively you can use the New-Volume cmdlet to achieve a similar result in a single command.

REMARKS

To see the examples, type: "get-help New-StoragePool -examples".

For more information, type: "get-help New-StoragePool -detailed".

For technical information, type: "get-help New-StoragePool -full".

For online help, type: "get-help New-StoragePool -online"