

# Update PC VM Categories

---

## Objective

The script is designed to allow for bulk updates of Categories at the Prism Central level based on a list of vms or a vm name input filter. The idea being that it is a simple way to assign, or remove, categories at scale for the likes of Citrix MCS or Citrix PVS provisioned workloads.

## Technical requirements for running the script

The script is compatible with Windows PowerShell 5.1 onwards.

This means that the technical requirements for the workstation or server running the script are as follows:

1. Any Windows version which can run Windows PowerShell 5.1
2. An appropriate credential with sufficient permissions to assign or remove categories to virtual machine entities within Prism Central

## Parameter Details

The following parameters exist to drive the behaviour of the script:

### Mandatory and recommended parameters:

- **pc\_source**: Mandatory **String**. The Prism Central Source to target.
- **Category**: Mandatory **String**. The name of the Category to assign or remove.
- **Value**: Mandatory **String**. The value of the Category to assign or remove.
- **IncludeList**: Optional. **array**. A list of machines to include in processing. Cannot be used with **VM\_Pattern\_Match**.
- **VM\_Pattern\_Match**: Optional **String**. A pattern match string to filter virtual machine entities. Cannot be used with **IncludeList**.
- **Mode**: Mandatory **String**. What mode to operate in, either **add** or **remove** for the Category assignment.

### Optional Parameters

- **ExclusionList**: Optional **array**. A list of vm names to exclude from processing.

- **LogPath**: Optional **String**. Log path output for all operations. The default is `C:\Logs\UpdatePCVMCategories.log`
- **LogRollover**: Optional **Int**. Number of days before log files are rolled over. Default is 5
- **VMCount**: Optional **Int**. The number of virtual machine entities to retrieve from PC. Defaults to 1000.
- **SleepTime**: Optional **Int**. The amount of time to sleep between API task retrieval. Defaults to 5 seconds.
- **APICallVerboseLogging**: Optional. **switch**. Switch to enable logging output for API calls.
- **UseCustomCredentialFile**: Optional. **switch**. Will call the `Get-CustomCredentials` function which keeps outputs and inputs a secure credential file base on Stephane Bourdeaud from Nutanix functions.
- **CredPath**: Optional **String**. Used if using the `UseCustomCredentialFile` parameter. Defines the location of the credential file. The default is `"$Env:USERPROFILE\Documents\WindowsPowerShell\CustomCredentials"`.
- **Whatif**: Optional. **Switch**. Will action the script in a whatif processing mode only.
- **async**: Optional. **Switch**. Will bypass a status check on each operation performed on the VM. This will decrease processing time significantly.

## Scenarios

Add a Category to all machines that match a naming input of Server1, Server2, and Server3 in a whatif processing mode

Param Splatting:

```
$params = @{
    pc_source      = "1.1.1.1"
    Category       = "Citrix_Provisioning_Type"
    Value          = "MCS"
    Mode           = "Add"
    IncludeList    = "Server1","Server2","Server3"
    UseCustomCredentialFile = $true
    Whatif         = $true
}

& UpdatePCVMCategories.ps1 $params
```

The direct script invocation via the command line with defined arguments would be:

```
.\UpdatePCVMCategories.ps1 -pc_source 1.1.1.1 -Category Citrix_Provisioning_Type -Value MCS -Mode Add -IncludeList "Server1","Server2","Server3" -UseCustomCredentialFile -Whatif
```

The script will:

- Connect to the Prism Central source **1.1.1.1**.
- Check for existence of the Category **Citrix\_Provisioning\_Type:MCS** pair before proceeding.
- Pull all virtual machine entities and filter based on the **IncludeList** array.
- Action an **Add** of the Category to each VM in the list.
- Process in a **whatif** mode to identify which machines will be actioned. No changes to the environment will occur.

Add a Category to all machines that match a naming input of \*MCS in a whatif processing mode

Param Splatting:

```
$params = @{  
    pc_source           = "1.1.1.1"  
    Category            = "Citrix_Provisioning_Type"  
    Value               = "MCS"  
    Mode                = "Add"  
    VM_Pattern_Match    = "*MCS"  
    ExclusionList        = "Server1","Server2"  
    UseCustomCredentialFile = $true  
    Whatif              = $true  
}  
  
& UpdatePCVMCategories.ps1 $params
```

The direct script invocation via the command line with defined arguments would be:

```
.\UpdatePCVMCategories.ps1 -pc_source 1.1.1.1 -Category Citrix_Provisioning_Type -Value MCS -Mode Add -VM_Pattern_Match
"*MCS" -ExclusionList "Server1","Server2" -UseCustomCredentialFile -Whatif
```

The script will:

- Connect to the Prism Central source **1.1.1.1**.
- Check for existence of the Category **Citrix\_Provisioning\_Type:MCS** pair before proceeding.
- Pull all virtual machine entities and filter based on the **\*MCS** name pattern.
- Exclude VMs matching either **Server1** or **Server2**.
- Action an **Add** of the Category to each VM in the list.
- Process in a **whatif** mode to identify which machines will be actioned. No changes to the environment will occur.

Add a Category to all machines that match a naming input of \*MCS

Param Splatting:

```
$params = @{
    pc_source          = "1.1.1.1"
    Category           = "Citrix_Provisioning_Type"
    Value              = "MCS"
    Mode               = "Add"
    VM_Pattern_Match   = "*MCS"
    UseCustomCredentialFile = $true
    async              = $true
}

& UpdatePCVMCategories.ps1 $params
```

The direct script invocation via the command line with defined arguments would be:

```
.\UpdatePCVMCategories.ps1 -pc_source 1.1.1.1 -Category Citrix_Provisioning_Type -Value MCS -Mode Add -VM_Pattern_Match
"*MCS" -UseCustomCredentialFile -async
```

The script will:

- Connect to the Prism Central source **1.1.1.1**.
- Check for existence of the Category **Citrix\_Provisioning\_Type:MCS** pair before proceeding.
- Pull all virtual machine entities and filter based on the **\*MCS** name pattern.
- Action an **Add** of the Category to each VM in the list.
- Operate in an async mode where category updates will not have their task status checked.

Remove a Category to all machines matching a naming input of Server1, Server2, and Server3

Param Splatting:

```
$params = @{
    pc_source           = "1.1.1.1"
    Category            = "Citrix_Provisioning_Type"
    Value              = "MCS"
    Mode               = "Remove"
    IncludeList         = "Server1","Server2","Server3"
    UseCustomCredentialFile = $true
    async              = $false
}

& UpdatePCVMCategories.ps1 $params
```

The direct script invocation via the command line with defined arguments would be:

```
.\UpdatePCVMCategories.ps1 -pc_source 1.1.1.1 -Category Citrix_Provisioning_Type -Value MCS -Mode Add -IncludeList
"Server1","Server2","Server3" -UseCustomCredentialFile
```

The script will:

- Connect to the Prism Central source **1.1.1.1**.
- Check for existence of the Category **Citrix\_Provisioning\_Type:MCS** pair before proceeding.
- Pull all virtual machine entities and filter based on the **IncludeList** array.
- Action a **Remove** of the Category to each VM in the list.
- Check each category action for a successful finish (task queried)

Remove a Category to all machines that match a naming input of \*MCS

Param Splatting:

```
$params = @{  
    pc_source           = "1.1.1.1"  
    Category            = "Citrix_Provisioning_Type"  
    Value              = "MCS"  
    Mode               = "Remove"  
    VM_Pattern_Match    = "*MCS"  
    UseCustomCredentialFile = $true  
    async              = $false  
}  
  
& UpdatePCVMCategories.ps1 $params
```

The direct script invocation via the command line with defined arguments would be:

```
.\UpdatePCVMCategories.ps1 -pc_source 1.1.1.1 -Category Citrix_Provisioning_Type -Value MCS -Mode Add -VM_Pattern_Match  
"*MCS" -UseCustomCredentialFile
```

The script will:

- Connect to the Prism Central source **1.1.1.1**.
- Check for existence of the Category **Citrix\_Provisioning\_Type:MCS** pair before proceeding.
- Pull all virtual machine entities and filter based on the **\*MCS** name pattern.
- Action a **Remove** of the Category to each VM in the list.
- Check each category action for a successful finish (task queried)

Add a Category to all machines that match a naming input of \*PVS

Param Splatting:

```
$params = @{  
    pc_source          = "1.1.1.1"  
    Category           = "Citrix_Provisioning_Type"  
    Value              = "PVS"  
    Mode               = "Add"  
    VM_Pattern_Match   = "*PVS"  
    UseCustomCredentialFile = $true  
}  
  
& UpdatePCVMCategories.ps1 $params
```

The direct script invocation via the command line with defined arguments would be:

```
.\UpdatePCVMCategories.ps1 -pc_source 1.1.1.1 -Category Citrix_Provisioning_Type -Value PVS -Mode Add -VM_Pattern_Match  
"*PVS" -UseCustomCredentialFile
```

The script will:

- Connect to the Prism Central source **1.1.1.1**.

- Check for existence of the Category `Citrix_Provisioning_Type:PVS` pair before proceeding.
- Pull all virtual machine entities and filter based on the `*PVS` name pattern.
- Action an `Add` of the Category to each VM in the list.
- Check each category action for a successful finish (task queried)