Step 02 -- Configuring Multipath-IO for Windows Server 2012/2012R2/2016/2019

Overview

There are three Windows Server management tasks that need to be completed for the Pure Storage FlashArray volumes (LUNs) to work properly. The first two discussed in the Configuring Multipath-IO section are:

1. Setup an MPIO Device
2. Configuring MPIO Timers

The next sections walkthrough each of these management tasks using the graphical user interface (GUI) or Windows PowerShell.

Multipath-IO Devices

The Pure Storage FlashArray needs to be setup as an MPIO Device in order to support multipathing. See the Installing Multipath-IO section for more details.

Setting up MPIO using the Control Panel Applet

These steps work for Windows Server 2012, 2012 R2, 2016, and 2019. The only differences between the graphical user interface is the changes in the dialogs visual appearances. The example provided in this section are taken from Windows Server 2016.

1. Using the Control Panel double-click on the MPIO applet to open up the MPIO Properties dialog box.

   Note: The MPIO applet can also be started from Start > Run mpiocpl.
2. Click **Add** to show the **Add MPIO Support** dialog box.

**Note:** Pay close attention to the instructions in the dialog box for string formatting. There should be 4 extra spaces after PURE and 6 extra spaces after FlashArray.

---

3rd party Device Specific Modules (DSMs) will not claim Pure Storage FlashArray volumes. No 3rd party DSMs are supported. This includes EMC PowerPath, NetApp ONTAP DSM, HP 3PAR DSM or others.
3. Click **OK** to complete the addition of the MPIO device.

---

**At this point configuration for Fibre Channel (FC) on the Windows Server is completed.** If using iSCSI please continue to [Setting up MPIO with iSCSI Support using the Control Panel Applet](#).

---

**A Reboot Required** dialog box will be displayed on all Windows Server versions. Choose **Yes** or **No** depending on what other management or application tasks you are performing but keep in mind that a reboot is required for the new settings to take effect.

**Important iSCSI Setup Note**

If iSCSI connectivity is being used **ignore** this reboot requirement. The reason for this is to reduce the number of reboot cycles for the Windows Server host because adding iSCSI support requires an additional reboot. If iSCSI is not being planned **Reboot** the Windows Server host.

---

### Setting up MPIO with iSCSI Support using the Control Panel Applet

1. **iSCSI Setup Step**

   Click on the **Discover Multi-Paths** tab in the **MPIO Properties** dialog. Click on **Add support for iSCSI devices**.
After adding the support for iSCSI devices click the **MPIO Devices** tab and see the **MSFT2005SCSIBusType_0x9** device has been added.
2. Click OK then Reboot the Windows Server host.

Setting up MPIO using Windows PowerShell on Windows Server 2012, 2012 R2, 2016 and 2019

Using Windows PowerShell is the preferred method for setting up MPIO devices. The reason for preferring PowerShell is the requirements of ensuring the device that is added adheres to the string formatting of Vendor (8 characters) and Product (16 characters). The PowerShell cmdlet, `New-MSDSMSupportedHw`, handles this formatting requirement.

The PowerShell below shows how to add a new MPIO device for Pure Storage, retrieve the MPIO devices and remove the example MPIO device provided by Windows Server.

Start a Windows PowerShell session and run the following.

**Add New MPIO Device**

```
PS C:\> New-MSDSMSupportedHw -VendorId PURE -ProductId FlashArray
```
Retrieve Available MPIO Devices

```
PS C:\> Get-MSDSMSupportedHw
VendorId ProductId
-------- ---------
Vendor 8 Product 16
PURE     FlashArray
```

Remove Default MPIO Device

Removing the default MPIO device is not necessary but simply a recommendation as the default Vendor 8 Product 16 device does not do anything.

```
PS C:\> Remove-MSDSMSupportedHw -VendorId 'Vendor*' -ProductId 'Product**'
PS C:\> Get-MSDSMSupportedHw
VendorId ProductId
-------- ---------
PURE     FlashArray
```

The final step to perform using PowerShell is to restart the computer. This operation is performed automatically when using the graphical user interface (GUI). In the case of PowerShell a command needs to be executed.

```
Restart-Computer
```

Multipath-IO Timers

There are 6 MPIO Timer values that are the recommended for use with a Pure Storage FlashArray for optimal performance. See Multipath-IO Timers for full details.

⚠️ When changing the Multilpath-IO timers a Reboot is Required for the new settings to take effect.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default</th>
<th>FlashArray</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomPathRecovery</td>
<td>0</td>
<td>1</td>
<td>Specifies whether MPIO performs custom path recovery.</td>
</tr>
</tbody>
</table>
### NewPathRecoveryInterval

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>Specifies a custom path recovery time, in seconds. This is the length of time before the server attempts path recovery. The default value is 40. <strong>Note:</strong> CustomPathRecovery parameter has to be enabled for this value to be used.</td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

### PDORemovePeriod

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Specifies a physical device object (PDO) removal period, in seconds. This period is the length of time the server waits after all paths to a PDO have failed before it removes the PDO. The default value is 20.</td>
</tr>
<tr>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

### NewDiskTimeout

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>Specifies the disk timeout value, in seconds. This value is the length of time the server waits before it marks the I/O request as timed out.</td>
</tr>
<tr>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>
Note: Microsoft documentation has an error and states the default `DiskTimeoutValue` is 120. On a newly installed Windows Server using `Get-MPIOSetting` shows the default value is 60. This is an error in Microsoft's documentation and this value should not be changed.

### Path Verification State

<table>
<thead>
<tr>
<th>Path Verification State</th>
<th>Disabled</th>
<th>Enabled</th>
</tr>
</thead>
</table>

### Path Verification Period

| Path Verification Period | 30 | 30 |

Specifies whether to enable path verification.

Specifies a path verification period, in seconds. This is the length of time for the server to verify every path. This parameter is not relevant unless the path verification state has a value of Enabled.

### Updating MPIO Timers using Windows PowerShell

Using Windows PowerShell is the preferred method for setting the MPIO Timer values when managing Windows Server 2012, 2012 R2, 2016 or 2019 using `Set-MPIOSetting` cmdlet which is part of the MPIO module.

Start a Windows PowerShell session and run the following.

#### Retrieving Current MPIO Timer Values

This will return the current MPIO Timer values. On a newly installed Windows Server all of the default settings will be set as shown below.

```
PS C:\> Get-MPIOSetting
PathVerificationState : Disabled
PathVerificationPeriod : 30
PDORemovePeriod      : 20
```
Update MPIO Timer Values

The below PowerShell shows running `Set-MPIOSetting` four different times with new parameter values. This was done to show each new timer value for clarity. The same can be accomplished with a single line of PowerShell using each of the parameters, this alternative is shown as well.

```
Set-MPIOSetting -NewPathRecoveryInterval 20
Set-MPIOSetting -CustomPathRecovery Enabled
Set-MPIOSetting -NewPDORemovePeriod 30
Set-MPIOSetting -NewDiskTimeout 60
Set-MPIOSetting -NewPathVerificationState Enabled

OR

Set-MPIOSetting -NewPathRecoveryInterval 20 -CustomPathRecovery Enabled -NewPDORemovePeriod 30 -NewDiskTimeout 60 -NewPathVerificationState Enabled
```