# Storage-Related Policies


- Configuring vHBA Templates, on page 1
- Configuring Fibre Channel Adapter Policies, on page 3
- Configuring the Default vHBA Behavior Policy, on page 9
- Configuring SAN Connectivity Policies, on page 11


# Configuring vHBA Templates

## vHBA Template

This template is a policy that defines how a vHBA on a server connects to the SAN. It is also referred to as a vHBA SAN connectivity template.

You must include this policy in a service profile for it to take effect.

## Configuring a vHBA Template

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A /org # **create vhba-templ** *vhba-templ-name* [**fabric** {**a** \| **b**}] [**fc-if** *vsan-name*] | Creates a vHBA template and enters organization vHBA template mode. |
| **Step 3** | (Optional) UCS-A /org/vhba-templ # **set descr** *description* | Provides a description for the vHBA template. |
| **Step 4** | (Optional) UCS-A /org/vhba-templ # **set fabric** {**a** \| **b**} | Specifies the fabric to use for the vHBA. If you did not specify the fabric when creating the vHBA template in Step 2, then you have the option to specify it with this command. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | (Optional) UCS-A /org/vhba-templ # **set fc-if** *vsan-name* | Specifies the Fibre Channel interface (named VSAN) to use for the vHBA template. If you did not specify the Fibre Channel interface when creating the vHBA template in Step 2, you have the option to specify it with this command. |
| **Step 6** | UCS-A /org/vhba-templ # **set max-field-size** *size-num* | Specifies the maximum size of the Fibre Channel frame payload (in bytes) that the vHBA supports. |
| **Step 7** | UCS-A /org/vhba-templ # **set pin-group** *group-name* | Specifies the pin group to use for the vHBA template. |
| **Step 8** | UCS-A /org/vhba-templ # **set qos-policy** *mac-pool-name* | Specifies the QoS policy to use for the vHBA template. |
| **Step 9** | UCS-A /org/vhba-templ # **set stats-policy** *policy-name* | Specifies the server and server component statistics threshold policy to use for the vHBA template. |
| **Step 10** | UCS-A /org/vhba-templ # **set type** {**initial-template** \| **updating-template**} | Specifies the vHBA template update type. If you do not want vHBA instances created from this template to be automatically updated when the template is updated, use the **initial-template** keyword; otherwise, use the **updating-template** keyword to ensure that all vHBA instances are updated when the vHBA template is updated. |
| **Step 11** | UCS-A /org/vhba-templ # **set wwpn-pool** *pool-name* | Specifies the WWPN pool to use for the vHBA template. |
| **Step 12** | UCS-A /org/vhba-templ # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example configures a vHBA template and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create vhba template VhbaTempFoo
UCS-A /org/vhba-templ* # set descr "This is a vHBA template example."
UCS-A /org/vhba-templ* # set fabric a
UCS-A /org/vhba-templ* # set fc-if accounting
UCS-A /org/vhba-templ* # set max-field-size 2112
UCS-A /org/vhba-templ* # set pin-group FcPinGroup12
UCS-A /org/vhba-templ* # set qos-policy policy34foo
UCS-A /org/vhba-templ* # set stats-policy ServStatsPolicy
UCS-A /org/vhba-templ* # set type updating-template
UCS-A /org/vhba-templ* # set wwpn-pool SanPool7
UCS-A /org/vhba-templ* # commit-buffer
UCS-A /org/vhba-templ #
```

# Deleting a vHBA Template

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* . |
| **Step 2** | UCS-A /org # **delete vhba-templ** *vhba-templ-name* | Deletes the specified vHBA template. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes the vHBA template named VhbaTempFoo and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete vhba template VhbaTempFoo
UCS-A /org* # commit-buffer
UCS-A /org #
```

# Configuring Fibre Channel Adapter Policies

## Ethernet and Fibre Channel Adapter Policies

These policies govern the host-side behavior of the adapter, including how the adapter handles traffic. For example, you can use these policies to change default settings for the following:

- Queues
- Interrupt handling
- Performance enhancement
- RSS hash
- Failover in a cluster configuration with two fabric interconnects

**Note** For Fibre Channel adapter policies, the values displayed by Cisco UCS Manager may not match those displayed by applications such as QLogic SANsurfer. For example, the following values may result in an apparent mismatch between SANsurfer and Cisco UCS Manager:

- Max LUNs Per Target—SANsurfer supports a maximum of 256 LUNs. Cisco UCS Manager, 6.0(2b) release version supports a maximum of 16,384 (16K) LUNs per vHBA (FC-Initiator) on Cisco UCS VIC 1400, 14000, and 15000 series adapters when used with Linux operating systems. For Windows and ESXi operating systems, the maximum number of LUNs supported is 4,096 (4K).This parameter is applicable only for FC-Initiator.
- Link Down Timeout—In SANsurfer, you configure the timeout threshold for link down in seconds. In Cisco UCS Manager, you configure this value in milliseconds. Therefore, a value of 5500 ms in Cisco UCS Manager displays as 5s in SANsurfer.
- Max Data Field Size—SANsurfer has allowed values of 512, 1024, and 2048. Cisco UCS Manager allows you to set values of any size. Therefore, a value of 900 in Cisco UCS Manager displays as 512 in SANsurfer.
- LUN Queue Depth—The LUN queue depth setting is available for Windows system FC adapter policies. Queue depth is the number of commands that the HBA can send and receive in a single transmission per LUN. Windows Storport driver sets this to a default value of 20 for physical miniports and to 250 for virtual miniports. This setting adjusts the initial queue depth for all LUNs on the adapter. Valid range for this value is 1 - 254. The default LUN queue depth is 20. This feature only works with Cisco UCS Manager version 3.1(2) and higher. This parameter is applicable only for FC-Initiator.
- IO TimeOut Retry—When the target device does not respond to an IO request within the specified timeout, the FC adapter cancels the pending command then resends the same IO after the timer expires. The FC adapter valid range for this value is 1 - 59 seconds. The default IO retry timeout is 5 seconds. This feature only works with Cisco UCS Manager version 3.1(2) and higher.

From Cisco UCS Manager 4.3(4a), the adapter settings are optimized for Windows, Linux, and VMware for Cisco UCS VIC 1400 Series adapters, Cisco UCS VIC 14000 Series adapters, and Cisco UCS VIC 15000 Series adapters.

### Operating System Specific Adapter Policies

By default, Cisco UCS provides a set of Ethernet adapter policies and Fibre Channel adapter policies. These policies include the recommended settings for each supported server operating system. Operating systems are sensitive to the settings in these policies. Storage vendors typically require non-default adapter settings. You can find the details of these required settings on the support list provided by those vendors.

**Important** We recommend that you use the values in these policies for the applicable operating system. Do not modify any of the values in the default policies unless directed to do so by Cisco Technical Support.

However, if you are creating an Ethernet adapter policy for an OS (instead of using the default adapter policy), you must use the following formulas to calculate values that work for that OS.

Depending on the UCS firmware, your driver interrupt calculations may be different. Newer UCS firmware uses a calculation that differs from previous versions. Later driver release versions on Linux operating systems now use a different formula to calculate the Interrupt Count. In this formula, the Interrupt Count is the maximum of either the Transmit Queue or the Receive Queue plus 2.

### Interrupt Count in Linux Adapter Policies

Drivers on Linux operating systems use differing formulas to calculate the Interrupt Count, depending on the eNIC driver version. The UCS 3.2 release increased the number of Tx and Rx queues for the eNIC driver from 8 to 256 each.

Use one of the following strategies, according to your driver version.

For Linux drivers before the UCS 3.2 firmware release, use the following formula to calculate the Interrupt Count.

Completion Queues = Transmit Queues + Receive Queues
Interrupt Count = (Completion Queues + 2) rounded up to nearest power of 2

For example, if Transmit Queues = 1 and Receive Queues = 8 then:

Completion Queues = 1 + 8 = 9
Interrupt Count = (9 + 2) rounded up to the nearest power of 2 = 16

On drivers for UCS firmware release 3.2 and higher, the Linux eNIC drivers use the following formula to calculate the Interrupt Count.

Interrupt Count = Max(Tx, Rx) + 2

For example:

Interrupt Count wq = 32, rq = 32, cq = 64 - then Interrupt Count = Max(32, 32) + 2 = 34
Interrupt Count wq = 64, rq = 8, cq = 72 – then Interrupt Count = Max(64, 8) + 2 = 66
Interrupt Count wq = 1, rq = 16, cq = 17 - then Interrupt count = Max(1, 16) + 2 = 18

### Interrupt Count in Windows Adapter Policies

For Windows OS, the recommended adapter policy in UCS Manager for VIC 1400 series and above adapters is Win-HPN and if RDMA is used, the recommended policy is Win-HPN-SMB. For VIC 1400 series and above adapters, the recommended interrupt value setting is 512 and the Windows VIC driver takes care of allocating the required number of Interrupts.

For VIC 1300 and VIC 1200 series adapters, the recommended UCS Manager adapter policy is Windows and the Interrupt would be TX + RX + 2, rounded to closest power of 2. The maximum supported Windows queues is 8 for Rx Queues and 1 for Tx Queues.

Example for VIC 1200 and VIC 1300 series adapters:

Tx = 1, Rx = 4, CQ = 5, Interrupt = 8 ( 1 + 4 rounded to nearest power of 2), Enable RSS

Example for VIC 1400 series , 14000 series and 15000 series adapters and above adapters:

Tx = 1, Rx = 4, CQ = 5, Interrupt = 512 , Enable RSS

### NVMe over Fabrics using Fibre Channel

The NVM Express (NVMe) interface allows host software to communicate with a non-volatile memory subsystem. This interface is optimized for Enterprise non-volatile storage, which is typically attached as a register level interface to the PCI Express (PCIe) interface.

NVMe over Fabrics using Fibre Channel (FC-NVMe) defines a mapping protocol for applying the NVMe interface to Fibre Channel. This protocol defines how Fibre Channel services and specified Information Units (IUs) are used to perform the services defined by NVMe over a Fibre Channel fabric. NVMe initiators can access and transfer information to NVMe targets over Fibre Channel.

FC-NVMe combines the advantages of Fibre Channel and NVMe. You get the improved performance of NVMe along with the flexibility and the scalability of the shared storage architecture. Cisco UCS Manager Release 4.0(2) supports NVMe over Fabrics using Fibre Channel on UCS VIC 1400 Series adapters.

Starting with UCS Manager release 4.3(2b), NVMeoF using RDMA is supported on Cisco UCS VIC 14000 series adapters.

Starting with UCS Manager release 4.2(2), NVMeoF using Fibre Channel is supported on Cisco UCS VIC 15000 series adapters.

Cisco UCS Manager provides the recommended FC NVME Initiator adapter policies in the list of pre-configured adapter policies. To create a new FC-NVMe adapter policy, follow the steps in the *Creating a Fibre Channel Adapter Policy* section.

### NVMe over Fabrics Using RDMA

NVMe over Fabrics (NVMeoF) is a communication protocol that allows one computer to access NVMe namespaces available on another computer. NVMeoF is similar to NVMe, but differs in the network-related steps involved in using the NVMeoF storage devices. The commands for discovering, connecting, and disconnecting a NVMeoF storage device are integrated into the **nvme** utility provided in Linux..

The NVMeoF fabric that Cisco supports is RDMA over Converged Ethernet version 2 (RoCEv2). RoCEv2 is a fabric protocol that runs over UDP. It requires a no-drop policy.

The eNIC RDMA driver works in conjunction with the eNIC driver, which must be loaded first when configuring NVMeoF.

Cisco UCS Manager provides the default Linux-NVMe-RoCE adapter policy for creating NVMe RoCEv2 interfaces. Do not use the default Linux adapter policy. For complete information on configuring RoCEv2 over NVMeoF, refer to the *Cisco UCS Manager Configuration Guide for RDMA over Converged Ethernet (RoCE) v2*.

NVMeoF using RDMA is supported on M5 B-Series or C-Series Servers with Cisco UCS VIC 1400 Series adapters.

Starting with UCS Manager release 4.3(2b), NVMeOF using RDMA is supported on Cisco UCS VIC 14000 series adapters.

Starting with UCS Manager release 4.2(2), NVMeOF using RDMA is supported on Cisco UCS VIC 15000 series adapters.

# Configuring a Fibre Channel Adapter Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A /org # **create fc-policy** *policy-name* | Creates the specified Fibre Channel adapter policy and enters organization Fibre Channel policy mode. |
| **Step 3** | (Optional) UCS-A /org/fc-policy # **set descr** *description* | Provides a description for the policy.<br>**Note**<br>If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output. |
| **Step 4** | (Optional) UCS-A /org/fc-policy # **set error-recovery** {**fcp-error-recovery** {**disabled** \| **enabled**} \| **link-down-timeout** *timeout-msec* \| **port-down-io-retry-count** *retry-count* \| **port-down-timeout** *timeout-msec*} | Configures the Fibre Channel error recovery. |
| **Step 5** | (Optional) UCS-A /org/fc-policy # **set interrupt mode** {**intx** \| **msi** \| **msi-x**}} | Configures the driver interrupt mode. |
| **Step 6** | (Optional) UCS-A /org/fc-policy # **set port** {**io-throttle-count** *throttle-count* \| **max-luns** *max-num*} | Configures the Fibre Channel port.<br>The maximum number of LUNs is usually controlled by the operating system running on the server. We recommend you consult your operating system (OS) documentation for the optimal value for this parameter.<br>• **Linux OS:** Enter an integer between 1 and 16384.<br>• **Windows OS and ESXi OS:** Enter an integer between 1 and 4096.<br>**Note**<br>The **max-luns** option is applicable only to the **fc-initiator** vHBA type. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | (Optional) UCS-A /org/fc-policy # **set port-f-logi** {**retries** *retry-count* \| **timeout** *timeout-msec*} | Configures the Fibre Channel port fabric login (FLOGI). |
| **Step 8** | (Optional) UCS-A /org/fc-policy # **set port-p-logi** {**retries** *retry-count* \| **timeout** *timeout-msec*} | Configures the Fibre Channel port-to-port login (PLOGI). |
| **Step 9** | (Optional) UCS-A /org/fc-policy # **set recv-queue** {**count** *count* \| **ring-size** *size-num*} | Configures the Fibre Channel receive queue. |
| **Step 10** | (Optional) UCS-A /org/fc-policy # **set scsi-io** {**count** *count* \| **ring-size** *size-num*} | Configures the Fibre Channel I/O. |
| **Step 11** | (Optional) UCS-A /org/fc-policy # **set trans-queue ring-size** *size-num*} | Configures the Fibre Channel transmit queue. |
| **Step 12** | (Optional) UCS-A /org/fc-policy # **set vhbatype mode** {**fc-initiator** \| **fc-nvme-initiator** \| **fc-nvme-target** \| **fc-target**} | The vHBA type used in this policy. vHBAs supporting FC and FC-NVMe can now be created on the same adapter.<br>**Note**<br>**fc-nvme-target** and **fc-target** are available as Tech Preview options. |
| **Step 13** | UCS-A /org/fc-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example configures a Fibre Channel adapter policy and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create fc-policy FcPolicy42
UCS-A /org/fc-policy* # set descr "This is a Fibre Channel adapter policy example."
UCS-A /org/fc-policy* # set error-recovery error-detect-timeout 2500
UCS-A /org/fc-policy* # set port max-luns 4
UCS-A /org/fc-policy* # set port-f-logi retries 250
UCS-A /org/fc-policy* # set port-p-logi timeout 5000
UCS-A /org/fc-policy* # set recv-queue count 1
UCS-A /org/fc-policy* # set scsi-io ring-size 256
UCS-A /org/fc-policy* # set trans-queue ring-size 256
UCS-A /org/fc-policy* # commit-buffer
UCS-A /org/fc-policy #
```

The following example configures a Fibre Channel adapter policy with the vHBA type set to FC NVME Initiator and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create fc-policy FcPolicy42
UCS-A /org/fc-policy* # set descr "This is a Fibre Channel adapter policy example."
UCS-A /org/fc-policy* # set error-recovery error-detect-timeout 2500
UCS-A /org/fc-policy* # set port-f-logi retries 250
UCS-A /org/fc-policy* # set port-p-logi timeout 5000
UCS-A /org/fc-policy* # set recv-queue count 1
```

```
UCS-A /org/fc-policy* # set scsi-io ring-size 256
UCS-A /org/fc-policy* # set trans-queue ring-size 256
UCS-A /org/fc-policy* # set vhbatype mode fc-nvme-initiator
UCS-A /org/fc-policy* # commit-buffer
UCS-A /org/fc-policy #
```

# Deleting a Fibre Channel Adapter Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* . |
| **Step 2** | UCS-A /org # **delete fc-policy** *policy-name* | Deletes the specified Fibre Channel adapter policy. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes the Fibre Channel adapter policy named FcPolicy42 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete fc-policy FcPolicy42
UCS-A /org* # commit-buffer
UCS-A /org #
```

# Configuring the Default vHBA Behavior Policy

## Default vHBA Behavior Policy

Default vHBA behavior policy allow you to configure how vHBAs are created for a service profile. You can choose to create vHBAs manually, or you can allow them to be created automatically.

You can configure the default vHBA behavior policy to define how vHBAs are created. This can be one of the following:

- **None**—Cisco UCS Manager does not create default vHBAs for a service profile. All vHBAs must be explicitly created.
- **HW Inherit**—If a service profile requires vHBAs and none have been explicitly defined, Cisco UCS Manager creates the required vHBAs based on the adapter installed in the server associated with the service profile.

**Note** If you do not specify a default behavior policy for vHBAs, **none** is used by default.

# Configuring a Default vHBA Behavior Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org /** | Enters the root organization mode. |
| **Step 2** | UCS-A/org # **scope vhba-beh-policy** | Enters default vHBA behavior policy mode. |
| **Step 3** | UCS-A/org/vhba-beh-policy # **set action** {**hw-inherit** [**template_name** *name*] \| **none**} | Specifies the default vHBA behavior policy. This can be one of the following:<br>• **hw-inherit**—If a service profile requires vHBAs and none have been explicitly defined, Cisco UCS Manager creates the required vHBAs based on the adapter installed in the server associated with the service profile.<br>If you specify **hw-inherit**, you can also specify a vHBA template to create the vHBAs.<br>• **none**—Cisco UCS Manager does not create default vHBAs for a service profile. All vHBAs must be explicitly created. |
| **Step 4** | UCS-A/org/vhba-beh-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to set the default vHBA behavior policy to **hw-inherit**.

```
UCS-A # scope org /
UCS-A/org # scope vhba-beh-policy
UCS-A/org/vhba-beh-policy # set action hw-inherit
UCS-A/org/vhba-beh-policy* # commit-buffer
UCS-A/org/vhba-beh-policy #
```

# Configuring SAN Connectivity Policies

## About the LAN and SAN Connectivity Policies

Connectivity policies determine the connections and the network communication resources between the server and the LAN or SAN on the network. These policies use pools to assign MAC addresses, WWNs, and WWPNs to servers and to identify the vNICs and vHBAs that the servers use to communicate with the network.

**Note** We do not recommend that you use static IDs in connectivity policies, because these policies are included in service profiles and service profile templates and can be used to configure multiple servers.

## Privileges Required for LAN and SAN Connectivity Policies

Connectivity policies enable users without network or storage privileges to create and modify service profiles and service profile templates with network and storage connections. However, users must have the appropriate network and storage privileges to create connectivity policies.

### Privileges Required to Create Connectivity Policies

Connectivity policies require the same privileges as other network and storage configurations. For example, you must have at least one of the following privileges to create connectivity policies:

- admin—Can create LAN and SAN connectivity policies
- ls-server—Can create LAN and SAN connectivity policies
- ls-network—Can create LAN connectivity policies
- ls-storage—Can create SAN connectivity policies

### Privileges Required to Add Connectivity Policies to Service Profiles

After the connectivity policies have been created, a user with ls-compute privileges can include them in a service profile or service profile template. However, a user with only ls-compute privileges cannot create connectivity policies.

## Interactions between Service Profiles and Connectivity Policies

You can configure the LAN and SAN connectivity for a service profile through either of the following methods:

- LAN and SAN connectivity policies that are referenced in the service profile
- Local vNICs and vHBAs that are created in the service profile
- Local vNICs and a SAN connectivity policy
- Local vHBAs and a LAN connectivity policy

Cisco UCS maintains mutual exclusivity between connectivity policies and local vNIC and vHBA configuration in the service profile. You cannot have a combination of connectivity policies and locally created vNICs or vHBAs. When you include a LAN connectivity policy in a service profile, all existing vNIC configuration is erased, and when you include a SAN connectivity policy, all existing vHBA configuration in that service profile is erased.

# Creating a SAN Connectivity Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **create san-connectivity-policy** *policy-name* | Creates the specified SAN connectivity policy, and enters organization network control policy mode.<br>This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved. |
| **Step 3** | (Optional) UCS-A /org/lan-connectivity-policy # **set descr** *policy-name* | Adds a description to the policy. We recommend that you include information about where and how the policy should be used.<br>Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote). |
| **Step 4** | UCS-A /org/service-profile # **set identity** {**dynamic-uuid** {*uuid* \| **derived**} \| **dynamic-wwnn** {*wwnn* \| **derived**} \| **uuid-pool** *pool-name* \| **wwnn-pool** *pool-name*} | Specifies how the server acquires a UUID or WWNN. You can do one of the following:<br>• Create a unique UUID in the form *nnnnnnnn-nnnn-nnnn-nnnnnnnnnnnn*<br>• Derive the UUID from the one burned into the hardware at manufacture<br>• Use a UUID pool<br>• Create a unique WWNN in the form *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh*<br>• Derive the WWNN from one burned into the hardware at manufacture |

| | Command or Action | Purpose |
|---|---|---|
| | | • Use a WWNN pool |
| **Step 5** | UCS-A /org/lan-connectivity-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy* # set descr "SAN connectivity policy"
UCS-A /org/san-connectivity-policy* # set identity wwnn-pool SanPool7
UCS-A /org/san-connectivity-policy* # commit-buffer
UCS-A /org/san-connectivity-policy #
```

### What to do next

Add one or more vHBAs and/or initiator groups to this SAN connectivity policy.

# Deleting a SAN Connectivity Policy

If you delete a SAN connectivity policy that is included in a service profile, it also deletes all vHBAs from that service profile and disrupts SAN data traffic for the server associated with the service profile.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **delete san-connectivity-policy** *policy-name* | Deletes the specified SAN connectivity policy. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to delete a SAN connectivity policy named SanConnect52 from the root organization and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete san-connectivity-policy SanConnect52
UCS-A /org* # commit-buffer
UCS-A /org #
```

# Creating a vHBA for a SAN Connectivity Policy

If you are continuing from Creating a SAN Connectivity Policy, on page 12, begin this procedure at Step 3.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **scope san-connectivity-policy** *policy-name* | Enters SAN connectivity policy mode for the specified SAN connectivity policy. |
| **Step 3** | UCS-A /org/san-connectivity-policy # **create vhba** *vhba-name* [**fabric** {**a** \| **b**}] [**fc-if** *fc-if-name*] | Creates a vHBA for the specified SAN connectivity policy and enters vHBA mode.<br>This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved. |
| **Step 4** | UCS-A /org/san-connectivity-policy/vhba # **set adapter-policy** *policy-name* | Specifies the adapter policy to use for the vHBA. |
| **Step 5** | UCS-A /org/san-connectivity-policy/vhba # **set identity** {**dynamic-wwpn** {*wwpn* \| **derived**} \| **wwpn-pool** *wwn-pool-name*} | Specifies the WWPN for the vHBA.<br>You can set the storage identity using one of the following options:<br>• Create a unique WWPN in the form *hh*:*hh*:*hh*:*hh*:*hh*:*hh*:*hh*:*hh*.<br>You can specify a WWPN in the range from 20:00:00:00:00:00:00:00 to 20:FF:FF:FF:FF:FF:FF:FF or from 50:00:00:00:00:00:00:00 to 5F:FF:FF:FF:FF:FF:FF:FF.<br>If you want the WWPN to be compatible with Cisco MDS Fibre Channel switches, use the WWPN template **20:00:00:25:B5:XX:XX:XX**.<br>• Derive the WWPN from one burned into the hardware at manufacture.<br>• Assign a WWPN from a WWN pool. |
| **Step 6** | UCS-A /org/san-connectivity-policy/vhba # **set max-field-size** *size-num* | Specifies the maximum size of the Fibre Channel frame payload (in bytes) that the vHBA supports. |

| | Command or Action | Purpose |
|---|---|---|
| | | Enter an integer between 256 and 2112. The default is 2048. |
| **Step 7** | UCS-A /org/san-connectivity-policy/vhba # **set order** {*order-num* \| **unspecified**} | Specifies the PCI scan order for the vHBA. |
| **Step 8** | UCS-A /org/san-connectivity-policy/vhba # **set pers-bind** {**disabled** \| **enabled**} | Disables or enables persistent binding to Fibre Channel targets. |
| **Step 9** | UCS-A /org/san-connectivity-policy/vhba # **set pin-group** *group-name* | Specifies the SAN pin group to use for the vHBA. |
| **Step 10** | UCS-A /org/san-connectivity-policy/vhba # **set qos-policy** *policy-name* | Specifies the QoS policy to use for the vHBA. |
| **Step 11** | UCS-A /org/san-connectivity-policy/vhba # **set stats-policy** *policy-name* | Specifies the statistics threshold policy to use for the vHBA. |
| **Step 12** | UCS-A /org/san-connectivity-policy/vhba # **set template-name** *policy-name* | Specifies the vHBA template to use for the vHBA. If you choose to use a vHBA template for the vHBA, you must still complete all of the configuration not included in the vHBA template, including Steps 4, 7, and 8. |
| **Step 13** | UCS-A /org/san-connectivity-policy/vhba # **set vcon** {**1** \| **2** \| **3** \| **4** \| **any**} | Assigns the vHBA to one or all virtual network interface connections. |
| **Step 14** | UCS-A /org/san-connectivity-policy/vhba # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to configure a vHBA for a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy* # create vhba vhba3 fabric a
UCS-A /org/san-connectivity-policy/vhba* # set adapter-policy AdaptPol2
UCS-A /org/san-connectivity-policy/vhba* # set identity wwpn-pool SanPool7
UCS-A /org/san-connectivity-policy/vhba* # set max-field-size 2112
UCS-A /org/san-connectivity-policy/vhba* # set order 0
UCS-A /org/san-connectivity-policy/vhba* # set pers-bind enabled
UCS-A /org/san-connectivity-policy/vhba* # set pin-group FcPinGroup12
UCS-A /org/san-connectivity-policy/vhba* # set qos-policy QosPol5
UCS-A /org/san-connectivity-policy/vhba* # set stats-policy StatsPol2
UCS-A /org/san-connectivity-policy/vhba* # set template-name SanConnPol3
UCS-A /org/san-connectivity-policy/vhba* # set vcon any
UCS-A /org/san-connectivity-policy/vhba* # commit-buffer
UCS-A /org/san-connectivity-policy/vhba #
```

### What to do next

If desired, add another vHBA or an initiator group to the SAN connectivity policy. If not, include the policy in a service profile or service profile template.

# Deleting a vHBA from a SAN Connectivity Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **scope san-connectivity-policy** *policy-name* | Enters SAN connectivity policy mode for the specified SAN connectivity policy. |
| **Step 3** | UCS-A /org/san-connectivity-policy # **delete vHBA** *vhba-name* | Deletes the specified vHBA from the SAN connectivity policy. |
| **Step 4** | UCS-A /org/san-connectivity-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to delete a vHBA named vHBA3 from a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy # delete vHBA vHBA3
UCS-A /org/san-connectivity-policy* # commit-buffer
UCS-A /org/san-connectivity-policy #
```

# Creating an Initiator Group for a SAN Connectivity Policy

If you are continuing from Creating a SAN Connectivity Policy, on page 12, begin this procedure at Step 3.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **scope san-connectivity-policy** *policy-name* | Enters SAN connectivity policy mode for the specified SAN connectivity policy. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | UCS-A /org/san-connectivity-policy # **create initiator-group** *group-name* **fc** | Creates the specified initiator group for Fibre Channel zoning and enters initiator group mode.<br>This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved. |
| **Step 4** | UCS-A /org/san-connectivity-policy/initiator-group # **create initiator** *vhba-name* | Creates the specified vHBA initiator in the initiator group.<br>If desired, repeat this step to add a second vHBA initiator to the group. |
| **Step 5** | UCS-A /org/san-connectivity-policy/initiator-group # **set storage-connection-policy** *policy-name* | Associates the specified storage connection policy with the SAN connectivity policy.<br>**Note**<br>This step assumes that you want to associate an existing storage connection policy to associate with the SAN connectivity policy. If you do, continue with Step 10. If you want to create a local storage definition for this policy instead, continue with Step 6. |
| **Step 6** | UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def # **create storage-target** *wwpn* | Creates a storage target endpoint with the specified WWPN, and enters storage target mode. |
| **Step 7** | UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target # **set target-path** {**a** \| **b**} | Specifies which fabric interconnect is used for communications with the target endpoint. |
| **Step 8** | UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target # **set target-vsan** *vsan* | Specifies which VSAN is used for communications with the target endpoint. |
| **Step 9** | UCS-A /org/san-connectivity-policy/initiator-group # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to configure an initiator group named initGroupZone1 with two initiators for a a SAN connectivity policy named SanConnect242, configure a local storage connection policy definition named scPolicyZone1, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy # create initiator-group initGroupZone1 fc
```

```
UCS-A /org/san-connectivity-policy/initiator-group* # set zoning-type sist
UCS-A /org/san-connectivity-policy/initiator-group* # create initiator vhba1
UCS-A /org/san-connectivity-policy/initiator-group* # create initiator vhba2
UCS-A /org/san-connectivity-policy/initiator-group* # create storage-connection-def
scPolicyZone1
UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def* # create
storage-target
20:10:20:30:40:50:60:70
UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target*
# set
target-path a
UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target*
# set
target-vsan default
UCS-A /org/san-connectivity-policy/initiator-group* # commit-buffer
UCS-A /org/san-connectivity-policy/initiator-group #
```

### What to do next

If desired, add another initiator group or a vHBA to the SAN connectivity policy. If not, include the policy in a service profile or service profile template.

# Creating an SPDM Security Policy

## SPDM Security

Cisco UCS M6, M7, and M8 servers can contain mutable components that could provide vectors for attack against a device itself or use of a device to attack another device within the system. To defend against these attacks, the Security Protocol and Data Model (SPDM) Specification enables a secure transport implementation that challenges a device to prove its identity and the correctness of its mutable component configuration.

SPDM defines messages, data objects, and sequences for performing message exchanges between devices over a variety of transport and physical media. It orchestrates message exchanges between Baseboard Management Controllers (BMC) and end-point devices over a Management Component Transport Protocol (MCTP). Message exchanges include authentication of hardware identities accessing the BMC. The SPDM enables access to low-level security capabilities and operations by specifying a managed level for device authentication, firmware measurement, and certificate management. Endpoint devices are challenged to provide authentication. and BMC authenticates the endpoints and only allows access for trusted entities.

The UCS Manager optionally allows uploads of external security certificates to BMC. A maximum of 40 SPDM certificates is allowed, including native internal certificates. Once the limit is reached, no more certificates can be uploaded. User uploaded certificates can be deleted but internal/default certificates cannot.

A SPDM security policy allows you to specify one of three Security level settings. Security can be set at one of the three levels listed below:

- Full Security:

  This is the highest MCTP security setting. When you select this setting, a fault is generated when any endpoint authentication failure or firmware measurement failure is detected. A fault will also be generated if any of the endpoints do not support either endpoint authentication or firmware measurements.

- Partial Security (default):

When you select this setting, a fault is generated when any endpoint authentication failure or firmware measurement failure is detected. There will NOT be a fault generated when the endpoint doesn't support endpoint authentication or firmware measurements.

- No Security

  When you select this setting, there will NOT be a fault generated for any failure (either endpoint measurement or firmware measurement failures).

You can also upload the content of one or more external/device certificates into BMC. Using a SPDM policy allows you to change or delete security certificates or settings as desired. Certificates can be deleted or replaced when no longer needed.

Certificates are listed in all user interfaces on a system.

## SPDM Authentication

The Security Protocol and Data Model (SPDM) is used by the BMC for authentication with the storage controller. It requires that the storage controller firmware is secure booted as well as having a Broadcom certificate chain installed in the slot0. During a firmware update, the Broadcom firmware will retain the older measurements for the storage firmware until the OCR or host reboots. If device authentication fails, the firmware will allow only inventory related commands and no set operations can be performed.

## Creating a SPDM Security Policy

A Security Protocol and Data Model (SPDM) policy can be created to present security alert-level and certificate contents to BMC for authentication.

- UCS-A# **scope org**

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A /org # **create spdm-certificate-policy** *policy-name* | Creates a SPDM security certificate policy with the specified policy name, and enters organization SPDM certificate policy mode. |
| **Step 3** | UCS-A /org/spdm-certificate-policy* # **set fault-alert** {**full** \| **partial** \| **no**} | Configures the fault alert level for this policy. |
| **Step 4** | (Optional) UCS-A /org/spdm-certificate-policy* # **set descr** *description* | Provides a description for the SPDM security certificate policy.<br>**Note**<br>If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the |

| | Command or Action | Purpose |
|---|---|---|
| | | description field of any **show** command output. |
| **Step 5** | UCS-A /org/spdm-certificate-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows creating a policy called "test" using an alert level of Partial Security (fault generated when an endpoint authentication or firmware measurement failure is detected). The default policy owner is Local.

```
UCS-A-FI-A /org #create spdm-certificate-policy test
UCS-A-FI-A /org /spdm-certificate-policy* # set?
fault-alert - Configure fault alert setting
desc - Description of policy
policy-owner  - Change ownership of policies
UCS-A-FI-A /org /spdm-certificate-policy* # set fault-alert partial
UCS-A-FI-A /org/spdm-certificate-policy* #commit-buffer
UCS-A-FI-A /org/spdm-certificate-policy# show details

SPDM Certificate Profile:
Name: test
Fault Alert Setting: partial
Description:
Policy Owner: Local
```

### What to do next

Assign outside security certificates, if desired.

## Loading an Outside SPDM Security Certificate Policy

The SPDM allows you to download an outside security certificate.

### Before you begin

Create a SPDM security certificate policy.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A /org # **scope spdm-certificate-policy** | Enters SPDM security certificate policy mode. |
| **Step 2** | UCS-A org/spdm-certificate-policy# **create spdm-cert** *Certificate name* | Creates a SPDM security certificate policy for the specified external certificate,. |
| **Step 3** | UCS-A /org/spdm-certificate-policy* # **set** {*certificate* } | Specifying certificate prompts for the content of the outside certificate. The only supported certificate type is **pem**. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | UCS-A /org/spdm-certificate-policy # **commit-buffer** | Commits the transaction to the system configuration. |

The following example shows loading a certificate for Broadcom of type PEM.

### Example

```
UCS-A-FI-A /org/spdm-certificate-policy# create spdm-cert?
Name - Certificate name

UCS-A-FI-A /org/spdm-certificate-policy# create spdm-cert Broadcom
UCS-A-FI-A /org/spdm-certificate-policy/spdm-cert* # set?
certificate - Certificate content

UCS-A-FI-A /org/spdm-certificate-policy/spdm-cert* # set certificate
{enter certificate content}
UCS-A-FI-A /org/spdm-certificate-policy/spdm-cert* # commit-buffer
UCS-A-FI-A /org/spdm-certificate-policy/spdm-cert# show detail
SPDM Certificate:
Name: Broadcom
Certificate Type: pem
Certificate Content:
```

## Displaying the Security Policy Fault Alert Level

After the policy is created, you can check the alert level for the SPDM policy.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A /org/spdm-certificate-policy # **show fault-alert**<br>**Example:**<br>`UCS-A /server/cimc/spdm-certificate #show fault-alert` | The returned result shows that the setting for this SPDM policy is Partial, the default.<br>`SPDM Fault Alert Setting: Partial` |

## Viewing the Certificate Inventory

You can view what SPDM certificates have been uploaded and also request further details for a specified certificate.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server* | |
| **Step 2** | UCS-A/server # **scope cimc** *server* | |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | UCS-A/server/cimc # **scope spdm** *server* | |
| **Step 4** | UCS-A/server/cimc/spdm # **show certificate** | The returned result shows the certificate inventory. |
| **Step 5** | UCS-A/server/cimc/spdm # **show certificate** *certificate-id***detail**<br>**Example:**<br>`UCS-A /server/cimc/spdm-certificate #show certificate 3 detail`<br>`Certificate Information`<br>`Certificate Id : 3`<br>`Subject Country Code (C) : US`<br>`Subject State (ST) : Colorado`<br>`Subject Organization (O) : Broadcom Inc.`<br>`Subject Organization Unit(OU) : NA`<br>`Subject Common Name (CN) : NA`<br>`Issuer Country Code (C) : US`<br>`Issuer State (ST) : Colorado`<br>`Issuer City (L) : Colorado Springs`<br>`Issuer Organization (O) : Broadcom Inc.`<br>`Issuer Organization Unit(OU) : NA`<br>`Issuer Common Name (CN) : NA`<br>`Valid From : Oct 23 00:25:13 2019 GMT`<br>`Valid To : Apr 8 10:36:14 2021 GMT`<br>`UserUploaded : Yes`<br>`Certificate Content : <Certificate String>`<br>`Certificate Type : PEM` | The returned result shows the certificate ID, identifiers, and expiration date. |
| **Step 6** | UCS-A /org/spdm-certificate-policy/certificate # **show**<br>**Example:**<br>`SPDM Certificate:`<br>`Name SPDM Certificate Type`<br>`----------`<br>`cert1 Pem`<br>**Example:**<br>`UCS-A /server/cimc/spdm-certificate/certificate #up`<br>`UCS-A /server/cimc/spdm-certificate #show`<br>`SPDM Certificate Policy:`<br>`Name Fault Alert Setting` | The returned result shows the type of certificate details.<br>The returned result shows the fault alert setting. |

| | Command or Action | Purpose |
|---|---|---|
| | ----------<br>----------<br>Broadcom Full | |

## Deleting a SPDM Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the org-name. |
| **Step 2** | UCS-A /org # **delete spdm-certificate-policy** *policy-name* | Deletes the specified SPDM control policy. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

#### Example

The following example deletes a power control policy called VendorPolicy2 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete spdm-certificate-policy VendorPolicy2
UCS-A /org* # commit-buffer
UCS-A /org #
```

# Deleting an Initiator Group from a SAN Connectivity Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **scope san-connectivity-policy** *policy-name* | Enters SAN connectivity policy mode for the specified SAN connectivity policy. |
| **Step 3** | UCS-A /org/san-connectivity-policy # **delete initiator-group** *group-name* | Deletes the specified initiator group from the SAN connectivity policy. |
| **Step 4** | UCS-A /org/san-connectivity-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to delete an initiator group named initGroup3 from a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy # delete initiator-group initGroup3
UCS-A /org/san-connectivity-policy* # commit-buffer
UCS-A /org/san-connectivity-policy #
```

# Configuring an Aero Controller Storage Profile

## Autoconfiguration Mode for Storage Controllers

Autoconfiguration mode for storage controllers are supported on Cisco UCS C-Series Rack servers (M6, M7, and M8), B-Series Blade servers (M6), and X-Series Compute Nodes (M6, M7, and M8).

C-series M6 servers support PCIe SAS316-port storage controllers for Direct Attached Storage. Controllers support an Autoconfiguration mode in which the state of a newly inserted disk is automatically moved to the Unconfigured-Good state.

Because of this, you can choose whether or not to use Autoconfiguration by creating a Storage Profile and associating it with the server. The default is that the automatic configuration feature is disabled, which retains the drive state when the server is rebooted.

If Autoconfiguration is used, you must select a drive state from one of the following:

- Unconfigured-Good
- JBOD
- RAID0 (RAID0 WriteBack)

This is because the controller firmware changes the behavior of systemPD to EPD-PT. EPD-PT is internally a RAID0 volume without any drive DDF metadata. The controller stores the metadata for identifying it as a RAID0 volume. The EPD-PT drives are considered as JBOD drives so the drive status is reported as JBOD and online.

Controller supports the following models:

- UCSC-RAID-M6T
- UCSC-RAID-M6HD
- UCSC-RAID-M6SD
- UCSX-X10C-RAIDF
- UCSC-RAID-HP

The table below shows the behavior of Autoconfiguration in different scenarios.

| Autoconfig Mode | Reboot/OCR | Hotplug | User Action |
|---|---|---|---|
| Unconfigured-Good (OFF) | • All Unconfigured-Good drives remain Unconfigured-Good.<br>• All previously configured JBOD remain JBOD. | • Inserted drive remains Unconfigured-Good.<br>• JBOD from a different server remains Unconfigured-Good on this controller. | Disabling Autoconfig has no impact on the existing configuration<br>Any JBOD device remains as JBOD across controller boot.<br>Any Unconfigured-Good remains unconfiguredgood across controller boot. |
| JBOD | • All Unconfigured-Good are converted to JBOD. | Newly inserted unconfigured device is converted to JBOD. | All Unconfigured-Good drives (non-user created) on the controller while running Autoconfig is converted to JBOD.<br>User created Unconfigured-Good drive remains Unconfigured-Good until next reboot. During reboot Unconfigured-Good gets converted to JBOD. |
| RAID0 (RAID0 WriteBack) | • All Unconfigured-Good converted to RAID0 WriteBack. | Newly inserted unconfigured device is converted to RAID0 WriteBack. | All Unconfigured-Good drives (non-user created) on the controller while running Autoconfig is converted to RAID0 WriteBack.<br>User created Unconfigured-Good remains Unconfigured-Good across controller reboot.<br>Any RAID0 WriteBack device remains as RAID0 WriteBack across controller reboot. |

Selecting EPD-PT (JBOD) as the default configuration does not retain the Unconfigured-Good state across host reboot. The drive state can be retained by disabling the automatic configuration feature. If the Autoconfig option is used, the default automatic configuration will always mark a drive as Unconfigured-Good.

When Autoconfig is selected, then the drive is configured to the desired drive state, the JBOD and unconfigured drives will set the drive state accordingly on the next controller boot or OCR,

The following table shows sample use cases for different Autoconfig scenarios.

| Use Case Scenario | Autoconfig Option |
|---|---|
| Using the server for JBOD Only (for example: Hyper converged, Hadoop data node etc ) | JBOD |
| Using the server for RAID volume (for example: SAP HANA database) | Unconfigured-Good |
| Using the server for Mixed JBOD and RAID volume | Unconfigured-Good |
| Using the server for per drive RAID0 WriteBack (for example: Hadoop data node) | RAID0 WriteBack |

**Note** When using UCSX-X10C-RAIDF or UCSC-RAID-M6T controllers with Autoconfiguration Mode set to RAID0, drives in *Unconfigured Good* state may not transition to *Online* and RAID0 LUNs will not be created after a storage profile redeploy and server reboot. This is a known behavior and differs from UCSC-RAID-HP controllers, where drives transition to *Online* and LUNs are created as expected.

## Creating an Autoconfiguation Profile

You can include the storage Autoconfiguration (Auto Config) mode option in your storage profile and unconfigure it when no longer needed. Changes will take effect on the next system boot. Auto Config for storage is only available on Cisco UCS M6 servers with Aero controllers.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A/org# **scope storage-profile** *profile-name* | Enters the storage profile for the specified profile. |
| **Step 3** | UCS-A/org/storage-profile# **show detail expand** | Shows a detailed view of the Storage Profile. If Auto Config Mode has not been enabled for this storage profile, or no Aero controller is present, you should not see an entry for Auto Config Mode. If Auto Config is not configured, inserted devices will retain their state on system reboot. |
| **Step 4** | UCS-A/org/storage-profile# **set auto-config-mode** jbod \| raid-0 \| unconfigured-good \| unspecified | Enables Auto Config Mode and sets the disk configuration mode to the desired state. If no further parameters are specified, all inserted devices will be tagged as Unconfigured Good on reboot. Enter unconfigured if you wish to disable Auto Config mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | UCS-A/org/storage-profile# **commit-buffer** | Commits the transaction to the system configuration. |