Using the UCS Utilities Within the Ultra M Manager

This appendix describes the UCS facilities within the Ultra M Manager.

Overview

Cisco UCS server BIOS, MLOM, and CIMC software updates may be made available from time to time.

Utilities have been added to the Ultra M Manager software to simplify the process of upgrading the UCS server software (firmware) within the Ultra M solution.

These utilities are available through a script called *ultram_ucs_utils.py* located in the */opt/cisco/usp/ultram-health* directory. Refer to ultram_ucs_utils.py Help for more information on this script.

NOTES:

- This functionality is currently supported only with Ultra M deployments based on OSP 10 and that leverage the Hyper-Converged architecture.
- You should only upgrade your UCS server software to versions that have been validated for use within the Ultra M solution.

- All UCS servers within the Ultra M solution stack should be upgraded to the same firmware versions.
- Though it is highly recommended that all server upgrades be performed during a single maintenance window, it is possible to perform the upgrade across multiple maintenance windows based on Node type (e.g. Compute, OSD Compute, and Controller).

There are two upgrade scenarios:

- **Upgrading servers in an existing deployment.** In the scenario, the servers are already in use hosting the Ultra M solution stack. This upgrade procedure is designed to maintain the integrity of the stack.
  - Compute Nodes are upgraded in parallel.
  - OSD Compute Nodes are upgraded sequentially.
  - Controller Nodes are upgraded sequentially.
- **Upgrading bare metal servers.** In this scenario, the bare metal servers have not yet been deployed within the Ultra M solution stack. This upgrade procedure leverages the parallel upgrade capability within Ultra M Manager UCS utilities to upgrade the servers in parallel.

To use UItra M Manager UCS utilities to upgrade software for UCS servers in an existing deployment:

1. Perform Pre-Upgrade Preparation.
2. Shutdown the ESC VMs, on page 6.
3. Upgrade the Compute Node Server Software.
4. Upgrade the OSD Compute Node Server Software, on page 8.
5. Restart the UAS and ESC (VNFM) VMs, on page 11.
6. Upgrade the Controller Node Server Software, on page 11.
7. Upgrade Firmware on the OSP-D Server/Ultra M Manager Node, on page 16.

To use UItra M Manager UCS utilities to upgrade software for bare metal UCS servers:

1. Perform Pre-Upgrade Preparation.
2. Upgrade Firmware on UCS Bare Metal, on page 14.
3. Upgrade Firmware on the OSP-D Server/Ultra M Manager Node, on page 16.

Perform Pre-Upgrade Preparation

Prior to performing the actual UCS server software upgrade, you must perform the steps in this section to prepare your environment for the upgrade.

NOTES:

- These instructions assume that all hardware is fully installed, cabled, and operational.
- These instructions assume that the VIM Orchestrator and VIM have been successfully deployed.
- UCS server software is distributed separately from the USP software ISO.

To prepare your environment prior to upgrading the UCS server software:

1. Log on to the Ultra M Manager Node.
2. Create a directory called */var/www/html/firmwares* to contain the upgrade files.

   mkdir -p /var/www/html/firmwares
3. Download the UCS software ISO to the directory you just created.

   UCS software is available for download from https://software.cisco.com/download/type.html?mdfid=286281356&flowid=71443
4. Extract the *bios.cap* file.

   mkdir /tmp/UCSISO

   sudo mount -t iso9660 -o loop ucs-c240m4-huu-<*version*>**.iso UCSISO/**

   ```
   mount: /dev/loop2 is write-protected, mounting read-only
   ```

   cd UCSISO/

   ls

   ```
   EFI                    GETFW            isolinux  Release-Notes-DN2.txt  squashfs_img.md5
   tools.squashfs.enc
   firmware.squashfs.enc  huu-release.xml  LiveOS    squashfs_img.enc.md5   TOC_DELNORTE2.xml
     VIC_FIRMWARE
   ```

   cd GETFW/

   ls

   ```
   getfw  readme.txt
   ```

   mkdir -p /tmp/HUU
   sudo ./getfw -s /tmp/ucs-c240m4-huu-<*version*>**.iso -d /tmp/HUU**

   ```
   Nothing was selected hence getting only CIMC and BIOS
   FW/s available at '/tmp/HUU/ucs-c240m4-huu-<version>'
   ```

   cd /tmp/HUU/ucs-c240m4-huu-<*version*>**/bios/**

   ls

   ```
   bios.cap
   ```
5. Copy the *bios.cap* and *huu.iso* to the */var/www/html/firmwares/* directory.

   sudo cp bios.cap /var/www/html/firmwares/

   ls -lrt /var/www/html/firmwares/

   ```
   total 692228
   -rw-r--r--.  1 root root 692060160 Sep 28 22:43 ucs-c240m4-huu-<version>.iso
   -rwxr-xr-x.  1 root root  16779416 Sep 28 23:55 bios.cap
   ```
6. *Optional.* If it is not already installed, install the Ultra M Manager using the information and instructions in Install the Ultra M Manager RPM.
7. Navigate to the */opt/cisco/usp/ultram-manager* directory.

   cd /opt/cisco/usp/ultram-manager

   Once this step is completed, if you are upgrading UCS servers in an existing Ultra M solution stack, proceed to 8, on page 3. If you are upgrading bare metal UCS servers, proceed to 9, on page 4.
8. *Optional.* If you are upgrading software for UCS servers in an existing Ultra M solution stack, then create UCS server node list configuration files for each node type as shown in the following table.

Configuration File Name	File Contents
compute.cfg	A list of the CIMC IP addresses for all of the Compute Nodes.
osd_compute_0.cfg	The CIMC IP address of the primary OSD Compute Node (osd-compute-0).
osd_compute_1.cfg	The CIMC IP address of the second OSD Compute Node (osd-compute-1).
osd_compute_2.cfg	The CIMC IP address of the third OSD Compute Node (osd-compute-2).
controller_0.cfg	The CIMC IP address of the primary Controller Node (controller-0).
controller_1.cfg	The CIMC IP address of the second Controller Node (controller-1).
controller_2.cfg	The CIMC IP address of the third Controller Node (controller-2).

Note Each address must be preceded by a dash and a space ("-"). The following is an example of the required format:

```
- 192.100.0.9
- 192.100.0.10
- 192.100.0.11
- 192.100.0.12
```

Separate configuration files are required for each OSD Compute and Controller Node in order to maintain the integrity of the Ultra M solution stack throughout the upgrade process.

9 *Optional.* If you are upgrading software on bare metal UCS servers prior to deploying them as part of the Ultra M solution stack, then create a configuration file called *hosts.cfg* containing a list of the CIMC IP addresses for all of the servers to be used within the Ultra M solution stack except the OSP-D server/Ultra M Manager Node.

Note Each address must be preceded by a dash and a space (-). The following is an example of the required format:

```
- 192.100.0.9
- 192.100.0.10
- 192.100.0.11
- 192.100.0.12
```

10 Create a configuration file called *ospd.cfg* containing the CIMC IP address of the OSP-D Server/Ultra M Manager Node.

Note The address must be preceded by a dash and a space ("-"). The following is an example of the required format:

```
- 192.300.0.9
```

11 Validate your configuration files by performing a sample test of the script to pull existing firmware versions from all Controller, OSD Compute, and Compute Nodes in your Ultra M solution deployment.

./ultram_ucs_utils.py --cfg "*<config_file_name>*" **--login** *<cimc_username> <cimc_user_password>* **--status 'firmwares'**
The following is an example output for a *hosts.cfg* file with a single Compute Node (192.100.0.7):

```
2017-10-01 10:36:28,189 - Successfully logged out from the server: 192.100.0.7
2017-10-01 10:36:28,190 -
--------------------------------------------------------------------------------------------
Server IP      | Component                                    | Version
--------------------------------------------------------------------------------------------
192.100.0.7    | bios/fw-boot-loader                        | C240M4.3.0.3c.0.0831170228

               | mgmt/fw-boot-loader                          | 3.0(3e).36
               | mgmt/fw-system                               | 3.0(3e)
               | adaptor-MLOM/mgmt/fw-boot-loader             | 4.1(2d)
               | adaptor-MLOM/mgmt/fw-system                  | 4.1(3a)
               | board/storage-SAS-SLOT-HBA/fw-boot-loader    |
6.30.03.0_4.17.08.00_0xC6130202
               | board/storage-SAS-SLOT-HBA/fw-system         | 4.620.00-7259
               | sas-expander-1/mgmt/fw-system                | 65104100
               | Intel(R) I350 1 Gbps Network Controller      | 0x80000E75-1.810.8
               | Intel X520-DA2 10 Gbps 2 port NIC            | 0x800008A4-1.810.8
               | Intel X520-DA2 10 Gbps 2 port NIC            | 0x800008A4-1.810.8
               | UCS VIC 1227 10Gbps 2 port CNA SFP+          | 4.1(3a)
               | Cisco 12G SAS Modular Raid Controller        | 24.12.1-0203
--------------------------------------------------------------------------------------------
```

If you receive errors when executing the script, ensure that the CIMC username and password are correct. Additionally, verify that all of the IP addresses have been entered properly in the configuration files.

Note It is highly recommended that you save the data reported in the output for later reference and validation after performing the upgrades.

12 Take backups of the various configuration files, logs, and other relevant information using the information and instructions in the *Backing Up Deployment Information* appendix in the *Ultra Services Platform Deployment Automation Guide*.

13 Continue the upgrade process based on your deployment status.

- Proceed to Shutdown the ESC VMs, on page 6 if you are upgrading software for servers that were previously deployed as part of the Ultra M solution stack.
- Proceed to Upgrade Firmware on UCS Bare Metal, on page 14 if you are upgrading software for servers that have not yet been deployed as part of the Ultra M solution stack.

Shutdown the ESC VMs

The Cisco Elastic Services Controller (ESC) serves as the VNFM in Ultra M solution deployments. ESC is deployed on a redundant pair of VMs. These VMs must be shut down prior to performing software upgrades on the UCS servers in the solution deployment.

To shut down the ESC VMs:

1. Login to OSP-D and make sure to "su - stack" and "source stackrc".
2. Run Nova list to get the UUIDs of the ESC VMs.

   nova list --fields name,host,status | grep <*vnf_deployment_name*>
   Example output:

```
<--- SNIP --->
| b470cfeb-20c6-4168-99f2-1592502c2057 | vnf1-ESC-ESC-
0                                                   | tb5-ultram-osd-compute-2.localdomain |
 ACTIVE |
| 157d7bfb-1152-4138-b85f-79afa96ad97d | vnf1-ESC-ESC-
1                                                   | tb5-ultram-osd-compute-1.localdomain |
 ACTIVE |
<--- SNIP --->
```

3. Stop the standby ESC VM.

   nova stop <*standby_vm_uuid*>
4. Stop the active ESC VM.

   nova stop <*active_vm_uuid*>
5. Verify that the VMs have been shutoff.

   nova list --fields name,host,status | grep <*vnf_deployment_name*>
   Look for the entries pertaining to the ESC UUIDs.

   Example output:

```
<--- SNIP --->
| b470cfeb-20c6-4168-99f2-1592502c2057 | vnf1-ESC-ESC-
0                                                   | tb5-ultram-osd-compute-2.localdomain |
 SHUTOFF |
| 157d7bfb-1152-4138-b85f-79afa96ad97d | vnf1-ESC-ESC-
1                                                   | tb5-ultram-osd-compute-1.localdomain |
 SHUTOFF |
<--- SNIP --->
```

6. Proceed to Upgrade the Compute Node Server Software, on page 6.

Upgrade the Compute Node Server Software

NOTES:

- Ensure that the ESC VMs have been shutdown according to the procedure in Shutdown the ESC VMs, on page 6.
- This procedure assumes that you are already logged in to the Ultra M Manager Node.
- This procedure requires the *compute.cfg* file created as part of the procedure detailed in Perform Pre-Upgrade Preparation, on page 2.

- It is highly recommended that all Compute Nodes be upgraded using this process during a single maintenance window.

To upgrade the UCS server software on the Compute Nodes:

1. Upgrade the BIOS on the UCS server-based Compute Nodes.

   ./ultram_ucs_utils.py --cfg "compute.cfg" --login <*cimc_username*> <*cimc_user_password*> **--upgrade bios --server** <*ospd_server_cimc_ip_address*> **--timeout 30 --file /firmwares/bios.cap**
   Example output:

```
2017-09-29 09:15:48,753 - Updating BIOS firmware on all the servers
2017-09-29 09:15:48,753 - Logging on UCS Server: 192.100.0.7
2017-09-29 09:15:48,758 - No session found, creating one on server: 192.100.0.7
2017-09-29 09:15:50,194 - Login successful to server: 192.100.0.7
2017-09-29 09:16:13,269 - 192.100.0.7 => updating | Image Download (5 %), OK
2017-09-29 09:17:26,669 - 192.100.0.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:18:34,524 - 192.100.0.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:19:40,892 - 192.100.0.7 => Activating BIOS
2017-09-29 09:19:55,011 -
--------------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
--------------------------------------------------------------------------
192.100.0.7     | SUCCESS | NA                  | Status: success, Progress: Done, OK
```

Note The Compute Nodes are automatically powered down after this process leaving only the CIMC interface available.

2. Upgrade the UCS server using the Host Upgrade Utility (HUU).

   ./ultram_ucs_utils.py --cfg "compute.cfg" --login <*cimc_username*> <*cimc_user_password*> **--upgrade huu --server** <*ospd_server_cimc_ip_address*> **--file /firmwares/**<*ucs_huu_iso_filename*>
   If the HUU script times out before completing the upgrade, the process might still be running on the remote hosts. You can periodically check the upgrade process by entering:

   ./ultram_ucs_utils.py --cfg "compute.cfg" --login <*cimc_username*> <*cimc_user_password*> **--status huu-upgrade**
   Example output:

```
--------------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
--------------------------------------------------------------------------
192.100.0.7    | SUCCESS | 2017-10-20 07:10:11 | Update Complete  CIMC Completed, SasExpDN
 Completed, I350 Completed, X520 Completed, X520 Completed, 3108AB-8i Completed, UCS VIC
 1227 Completed, BIOS Completed,
--------------------------------------------------------------------------
```

3. Verify that the BIOS firmware and HUU upgrade was successful by checking the post-upgrade versions.

   ./ultram_ucs_utils.py --cfg "compute.cfg" --login <*cimc_username*> <*cimc_user_password*> **--status firmwares**

4. Set the package-c-state-limit CIMC setting.

   ./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg compute.cfg --login<*cimc_username*> <*cimc_user_password*>

5. Verify that the package-c-state-limit CIMC setting has been made.

   ./ultram_ucs_utils.py --status bios-settings --cfg compute.cfg --login <*cimc_username*> <*cimc_user_password*>
   Look for **PackageCStateLimit** to be set to *C0/C1*.

6. Modify the Grub configuration on each Compute Node.

a. Log into your first compute (compute-0) and update the grub setting with "processor.max_cstate=0 intel_idle.max_cstate=0".

sudo grubby --info=/boot/vmlinuz-`uname -r`
sudo grubby --update-kernel=/boot/vmlinuz-`uname -r` --args="processor.max_cstate=0 intel_idle.max_cstate=0"

b. Verify that the update was successful.

sudo grubby --info=/boot/vmlinuz-`uname -r`
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

c. Reboot the Compute Nodes.

sudo reboot

d. Repeat steps 6.a, on page 8 through 6.c, on page 8 for all other Compute Nodes.

7. Recheck all CIMC and kernel settings.

a. Log in to the Ultra M Manager Node.

b. Verify CIMC settings

./ultram_ucs_utils.py --status bios-settings --cfg compute.cfg --login *<cimc_username> <cimc_user_password>*

c. Verify the processor c-state.

for ip in `nova list | grep -i compute | awk '{print $12}' | sed 's/ctlplane=//g'`; do ssh heat-admin@$ip 'sudo cat /sys/module/intel_idle/parameters/max_cstate'; done
for ip in `nova list | grep -i compute | awk '{print $12}' | sed 's/ctlplane=//g'`; do ssh heat-admin@$ip 'sudo cpupower idle-info'; done

8. Proceed to Upgrade the OSD Compute Node Server Software.

Note Other Node types can be upgraded at a later time. If you'll be upgrading them during a later maintenance window, proceed to Restart the UAS and ESC (VNFM) VMs, on page 11.

Upgrade the OSD Compute Node Server Software

NOTES:

- This procedure requires the *osd_compute_0.cfg*, *osd_compute_1.cfg*, and *osd_compute_2.cfg* files created as part of the procedure detailed in Perform Pre-Upgrade Preparation, on page 2.
- It is highly recommended that all OSD Compute Nodes be upgraded using this process during a single maintenance window.

To upgrade the UCS server software on the OSD Compute Nodes:

1. Move the Ceph storage to maintenance mode.

a. Log on to the lead Controller Node (controller-0).

b Move the Ceph storage to maintenance mode.

sudo ceph status
sudo ceph osd set noout
sudo ceph osd set norebalance
sudo ceph status

2 *Optional.* If they've not already been shut down, shut down both ESC VMs using the instructions in Shutdown the ESC VMs, on page 6.

3 Log on to the Ultra M Manager Node.

4 Upgrade the BIOS on the initial UCS server-based OSD Compute Node (osd-compute-1).

./ultram_ucs_utils.py --cfg "osd_compute_0.cfg" --login <*cimc_username*> <*cimc_user_password*> **--upgrade bios --server** <*ospd_server_cimc_ip_address*> **--timeout 30 --file /firmwares/bios.cap**
Example output:

```
2017-09-29 09:15:48,753 - Updating BIOS firmware on all the servers
2017-09-29 09:15:48,753 - Logging on UCS Server: 192.100.0.17
2017-09-29 09:15:48,758 - No session found, creating one on server: 192.100.0.17
2017-09-29 09:15:50,194 - Login successful to server: 192.100.0.17
2017-09-29 09:16:13,269 - 192.100.0.17 => updating | Image Download (5 %), OK
2017-09-29 09:17:26,669 - 192.100.0.17 => updating | Write Host Flash (75 %), OK
2017-09-29 09:18:34,524 - 192.100.0.17 => updating | Write Host Flash (75 %), OK
2017-09-29 09:19:40,892 - 192.100.0.17 => Activating BIOS
2017-09-29 09:19:55,011 -
--------------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
--------------------------------------------------------------------------
192.100.0.17     | SUCCESS | NA                  | Status: success, Progress: Done, OK
```

Note The Compute Nodes are automatically powered down after this process leaving only the CIMC interface available.

5 Upgrade the UCS server using the Host Upgrade Utility (HUU).

./ultram_ucs_utils.py --cfg "osd_compute.cfg" --login <*cimc_username*> <*cimc_user_password*> **--upgrade huu --server** <*ospd_server_cimc_ip_address*> **--file /firmwares/**<*ucs_huu_iso_filename*>
If the HUU script times out before completing the upgrade, the process might still be running on the remote hosts. You can periodically check the upgrade process by entering:

./ultram_ucs_utils.py --cfg "osd_compute.cfg" --login <*cimc_username*> <*cimc_user_password*> **--status huu-upgrade**
Example output:

```
--------------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
--------------------------------------------------------------------------
192.100.0.17    | SUCCESS | 2017-10-20 07:10:11 | Update Complete  CIMC Completed, SasExpDN
 Completed, I350 Completed, X520 Completed, X520 Completed, 3108AB-8i Completed, UCS VIC
 1227 Completed, BIOS Completed,
--------------------------------------------------------------------------
```

6 Verify that the BIOS firmware and HUU upgrade was successful by checking the post-upgrade versions.

./ultram_ucs_utils.py --cfg "osd_compute_0.cfg" --login <*cimc_username*> <*cimc_user_password*> **--status firmwares**

7 Set the package-c-state-limit CIMC setting.

./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg osd_compute_0.cfg --login <*cimc_username*> <*cimc_user_password*>

8 Verify that the package-c-state-limit CIMC setting has been made.

./ultram_ucs_utils.py --status bios-settings --cfg osd_compute_0.cfg --login <*cimc_username*> <*cimc_user_password*>
Look for **PackageCStateLimit** to be set to *C0/C1*.

9 Modify the Grub configuration on the primary OSD Compute Node.

a Log on to the OSD Compute (osd-compute-0) and update the grub setting with "processor.max_cstate=0 intel_idle.max_cstate=0".

sudo grubby --info=/boot/vmlinuz-`uname -r`
sudo grubby --update-kernel=/boot/vmlinuz-`uname -r` --args="processor.max_cstate=0 intel_idle.max_cstate=0"

b Verify that the update was successful.

sudo grubby --info=/boot/vmlinuz-`uname -r`
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

c Reboot the OSD Compute Nodes.

sudo reboot

10 Recheck all CIMC and kernel settings.

a Verify the processor c-state.

cat /sys/module/intel_idle/parameters/max_cstate
cpupower idle-info

b Login to Ultra M Manager Node.

c Verify CIMC settings.

./ultram_ucs_utils.py --status bios-settings --cfg osd_compute_0.cfg --login <*cimc_username*> <*cimc_user_password*>

11 Repeat steps 4, on page 9 through 10, on page 10 on the second OSD Compute Node (osd-compute-1).

Note Be sure to use the *osd_compute_1.cfg* file where needed.

12 Repeat steps 4, on page 9 through 10, on page 10 on the third OSD Compute Node (osd-compute-2).

Note Be sure to use the *osd_compute_2.cfg* file where needed.

13 Check the ironic node-list and restore any hosts that went into maintenance mode true state.

a Login to OSP-D and make sure to "su - stack" and "source stackrc".

b Perform the check and any required restorations.

ironic node-list
ironic node-set-maintenance $NODE_<*node_uuid*> **off**

14 Move the Ceph storage out of maintenance mode.

a Log on to the lead Controller Node (controller-0).

b Move the Ceph storage to maintenance mode.

sudo ceph status
sudo ceph osd unset noout
sudo ceph osd unset norebalance
sudo ceph status
sudo pcs status

15 Proceed to Restart the UAS and ESC (VNFM) VMs, on page 11.

Restart the UAS and ESC (VNFM) VMs

Upon performing the UCS server software upgrades, VMs that were previously shutdown must be restarted.

To restart the VMs:

1 Login to OSP-D and make sure to "su - stack" and "source stackrc".

2 Run Nova list to get the UUIDs of the ESC VMs.

3 Start the AutoIT VM.

nova start <*autoit_vm_uuid*>

4 Start the AutoDeploy VM.

nova start <*autodeploy_vm_uuid*>

5 Start the standby ESC VM.

nova start <*standby_vm_uuid*>

6 Start the active ESC VM.

nova start <*active_vm_uuid*>

7 Verify that the VMs have been restarted and are ACTIVE.

nova list --fields name,host,status | grep <*vnf_deployment_name*>
Once ESC is up and running, it triggers the recovery of rest of the VMs (AutoVNF, UEMs, CFs and SFs).

8 Login to each of the VMs and verify that they are operational.

Upgrade the Controller Node Server Software

NOTES:

- This procedure requires the *controller_0.cfg*, *controller_1.cfg*, and *controller_2.cfg* files created as part of the procedure detailed in Perform Pre-Upgrade Preparation, on page 2.
- It is highly recommended that all Controller Nodes be upgraded using this process during a single maintenance window.

To upgrade the UCS server software on the Controller Nodes:

1 Check the Controller Node status and move the Pacemaker Cluster Stack (PCS) to maintenance mode.

a Login to the primary Controller Node (controller-0) from the OSP-D Server.

b Check the state of the Controller Node Pacemaker Cluster Stack (PCS).

sudo pcs status

Note Resolve any issues prior to proceeding to the next step.

c Place the PCS cluster on the Controller Node into standby mode.

sudo pcs cluster standby <*controller_name*>

d Recheck the Controller Node status again and make sure that the Controller Node is in standby mode for the PCS cluster.

sudo pcs status

2 Log on to the Ultra M Manager Node.

3 Upgrade the BIOS on the primary UCS server-based Controller Node (controller-0).

./ultram_ucs_utils.py --cfg "controller_0.cfg" --login <*cimc_username*> <*cimc_user_password*> **--upgrade bios --server** <*ospd_server_cimc_ip_address*> **--timeout 30 --file /firmwares/bios.cap**
Example output:

```
2017-09-29 09:15:48,753 - Updating BIOS firmware on all the servers
2017-09-29 09:15:48,753 - Logging on UCS Server: 192.100.2.7
2017-09-29 09:15:48,758 - No session found, creating one on server: 192.100.2.7
2017-09-29 09:15:50,194 - Login successful to server: 192.100.2.7
2017-09-29 09:16:13,269 - 192.100.2.7 => updating | Image Download (5 %), OK
2017-09-29 09:17:26,669 - 192.100.2.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:18:34,524 - 192.100.2.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:19:40,892 - 192.100.2.7 => Activating BIOS
2017-09-29 09:19:55,011 -
--------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
--------------------------------------------------------------------
192.100.2.7     | SUCCESS | NA                  | Status: success, Progress: Done, OK
```

Note The Compute Nodes are automatically powered down after this process leaving only the CIMC interface available.

4 Upgrade the UCS server using the Host Upgrade Utility (HUU).

./ultram_ucs_utils.py --cfg "controller_0.cfg" --login <*cimc_username*> <*cimc_user_password*> **--upgrade huu --server** <*ospd_server_cimc_ip_address*> **--file /firmwares/**<*ucs_huu_iso_filename*>
If the HUU script times out before completing the upgrade, the process might still be running on the remote hosts. You can periodically check the upgrade process by entering:

./ultram_ucs_utils.py --cfg "controller_0.cfg" --login <*cimc_username*> <*cimc_user_password*> **--status huu-upgrade**
Example output:

```
--------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
--------------------------------------------------------------------
192.100.2.7    | SUCCESS | 2017-10-20 07:10:11 | Update Complete  CIMC Completed, SasExpDN
 Completed, I350 Completed, X520 Completed, X520 Completed, 3108AB-8i Completed, UCS VIC
 1227 Completed, BIOS Completed,
--------------------------------------------------------------------
```

5 Verify that the BIOS firmware and HUU upgrade was successful by checking the post-upgrade versions.

./ultram_ucs_utils.py --cfg "controller_0.cfg" --login <*cimc_username*> <*cimc_user_password*> **--status firmwares**

6 Set the package-c-state-limit CIMC setting.

./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg controller_0.cfg --login <*cimc_username*> <*cimc_user_password*>

7 Verify that the package-c-state-limit CIMC setting has been made.

./ultram_ucs_utils.py --status bios-settings --cfg controller_0.cfg --login <*cimc_username*> <*cimc_user_password*>
Look for **PackageCStateLimit** to be set to *C0/C1*.

8 Modify the Grub configuration on the primary OSD Compute Node.

a Log on to the OSD Compute (osd-compute-0) and update the grub setting with "processor.max_cstate=0 intel_idle.max_cstate=0".

sudo grubby --info=/boot/vmlinuz-`uname -r`
sudo grubby --update-kernel=/boot/vmlinuz-`uname -r` --args="processor.max_cstate=0 intel_idle.max_cstate=0"

b Verify that the update was successful.

sudo grubby --info=/boot/vmlinuz-`uname -r`
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

c Reboot the OSD Compute Nodes.

sudo reboot

9 Recheck all CIMC and kernel settings.

a Verify the processor c-state.

cat /sys/module/intel_idle/parameters/max_cstate
cpupower idle-info

b Login to Ultra M Manager Node.

c Verify CIMC settings.

./ultram_ucs_utils.py --status bios-settings --cfg controller_0.cfg --login <*cimc_username*> <*cimc_user_password*>

10 Check the ironic node-list and restore the Controller Node if it went into maintenance mode true state.

a Login to OSP-D and make sure to "su - stack" and "source stackrc".

b Perform the check and any required restorations.

ironic node-list
ironic node-set-maintenance $NODE_<*node_uuid*> **off**

11 Take the Controller Node out of the PCS standby state.

sudo pcs cluster unstandby <*controller-0-id*>

12 Wait 5 to 10 minutes and check the state of the PCS cluster to verify that the Controller Node is ONLINE and all services are in good state.

sudo pcs status

13 Repeat steps 3, on page 12 through 11, on page 13 on the second Controller Node (controller-1).

Note Be sure to use the *controller_1.cfg* file where needed.

14 Repeat steps 3, on page 12 through 11, on page 13 on the third Controller Node (controller-2).

Note Be sure to use the *controller_2.cfg* file where needed.

15 Proceed to Upgrade Firmware on the OSP-D Server/Ultra M Manager Node, on page 16.

Upgrade Firmware on UCS Bare Metal

NOTES:

- This procedure assumes that the UCS servers receiving the software (firmware) upgrade have not previously been deployed as part of an Ultra M solution stack.
- The instructions in this section pertain to all servers to be used as part of an Ultra M solution stack except the OSP-D Server/Ultra M Manager Node.
- This procedure requires the *hosts.cfg* file created as part of the procedure detailed in Perform Pre-Upgrade Preparation, on page 2.

To upgrade the software on the UCS servers:

1 Log on to the Ultra M Manager Node.

2 Upgrade the BIOS on the UCS servers.

./ultram_ucs_utils.py --cfg "hosts.cfg" --login <*cimc_username*> <*cimc_user_password*> **--upgrade bios --server** <*ospd_server_cimc_ip_address*> **--timeout 30 --file /firmwares/bios.cap**
Example output:

```
2017-09-29 09:15:48,753 - Updating BIOS firmware on all the servers
2017-09-29 09:15:48,753 - Logging on UCS Server: 192.100.0.7
2017-09-29 09:15:48,758 - No session found, creating one on server: 192.100.0.7
2017-09-29 09:15:50,194 - Login successful to server: 192.100.0.7
2017-09-29 09:16:13,269 - 192.100.0.7 => updating | Image Download (5 %), OK
2017-09-29 09:17:26,669 - 192.100.0.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:18:34,524 - 192.100.0.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:19:40,892 - 192.100.0.7 => Activating BIOS
2017-09-29 09:19:55,011 -
--------------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
--------------------------------------------------------------------------
192.100.0.7     | SUCCESS | NA                  | Status: success, Progress: Done, OK
```

Note The Compute Nodes are automatically powered down after this process leaving only the CIMC interface available.

3 Upgrade the UCS server using the Host Upgrade Utility (HUU).

./ultram_ucs_utils.py --cfg "hosts.cfg" --login <*cimc_username*> <*cimc_user_password*> **--upgrade huu --server** <*ospd_server_cimc_ip_address*> **--file /firmwares/**<*ucs_huu_iso_filename*>
If the HUU script times out before completing the upgrade, the process might still be running on the remote hosts. You can periodically check the upgrade process by entering:

./ultram_ucs_utils.py --cfg "hosts.cfg" --login <*cimc_username*> <*cimc_user_password*> **--status huu-upgrade**

Example output:

```
--------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
--------------------------------------------------------------------
192.100.0.7    | SUCCESS | 2017-10-20 07:10:11 | Update Complete  CIMC Completed, SasExpDN
 Completed, I350 Completed, X520 Completed, X520 Completed, 3108AB-8i Completed, UCS VIC
 1227 Completed, BIOS Completed,
--------------------------------------------------------------------
```

4 Verify that the BIOS firmware and HUU upgrade was successful by checking the post-upgrade versions.

./ultram_ucs_utils.py --cfg "hosts.cfg" --login <*cimc_username*> <*cimc_user_password*> **--status firmwares**

5 Set the package-c-state-limit CIMC setting.

./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg hosts.cfg --login <*cimc_username*> <*cimc_user_password*>

6 Verify that the package-c-state-limit CIMC setting has been made.

./ultram_ucs_utils.py --status bios-settings --cfg hosts.cfg --login <*cimc_username*> <*cimc_user_password*>
Look for **PackageCStateLimit** to be set to *C0/C1*.

7 Recheck all CIMC and BIOS settings.

a Log in to the Ultra M Manager Node.

b Verify CIMC settings.

./ultram_ucs_utils.py --status bios-settings --cfg hosts.cfg --login <*cimc_username*> <*cimc_user_password*>

8 Modify the "ComputeKernelArgs" statement in the *network.yaml* with the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments.

vi network.yaml

```
<---SNIP--->
ComputeKernelArgs: "intel_iommu=on default_hugepagesz=1GB hugepagesz=1G hugepages=12
processor.max_cstate=0 intel_idle.max_cstate=0"
```

9 Modify the Grub configuration on all Controller Nodes after the VIM (Overcloud) has been deployed.

a Log into your first Controller Node (controller-0).

ssh heat-admin@<*controller_address*>

b Check the grubby settings.

sudo grubby --info=/boot/vmlinuz-`uname -r`
Example output:

```
index=0
kernel=/boot/vmlinuz-3.10.0-514.21.1.el7.x86_64
args="ro console=tty0 console=ttyS0,115200n8 crashkernel=auto rhgb quiet "
root=UUID=fa9e939e-9e3c-4f1c-a07c-3f506756ad7b
initrd=/boot/initramfs-3.10.0-514.21.1.el7.x86_64.img
title=Red Hat Enterprise Linux Server (3.10.0-514.21.1.el7.x86_64) 7.3 (Maipo)
```

c Update the grub setting with the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments.

sudo grubby --update-kernel=/boot/vmlinuz-`uname -r` --args="processor.max_cstate=0 intel_idle.max_cstate=0"

d Verify that the update was successful.

sudo grubby --info=/boot/vmlinuz-`uname -r`
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

Example output:

```
index=0
kernel=/boot/vmlinuz-3.10.0-514.21.1.el7.x86_64
args="ro console=tty0 console=ttyS0,115200n8 crashkernel=auto rhgb quiet
processor.max_cstate=0 intel_idle.max_cstate=0"
root=UUID=fa9e939e-9e3c-4f1c-a07c-3f506756ad7b
initrd=/boot/initramfs-3.10.0-514.21.1.el7.x86_64.img
title=Red Hat Enterprise Linux Server (3.10.0-514.21.1.el7.x86_64) 7.3 (Maipo)
```

e Reboot the Controller Node.

sudo reboot

Important Do not proceed with the next step until the Controller Node is up and rejoins the cluster.

f Repeat steps 9.a, on page 15 through 9.e, on page 16 for all other Controller Nodes.

10 Proceed to Upgrade Firmware on the OSP-D Server/Ultra M Manager Node, on page 16.

Upgrade Firmware on the OSP-D Server/Ultra M Manager Node

1 Open your web browser.

2 Enter the CIMC address of the OSP-D Server/Ultra M Manager Node in the URL field.

3 Login to the CIMC using the configured user credentials.

4 Click **Launch KVM Console**.

5 Click **Virtual Media**.

6 Click **Add Image** and select the HUU ISO file pertaining to the version you wish to upgrade to.

7 Select the ISO that you have added in the **Mapped** column of the **Client View**. Wait for the selected ISO to appear as a mapped device.

8 Boot the server and press F6 when prompted to open the **Boot Menu**.

9 Select the desired ISO.

10 Select **Cisco vKVM-Mapped vDVD1.22**, and press **Enter**. The server boots from the selected device.

11 Follow the onscreen instructions to update the desired software and reboot the server. Proceed to the next step once the server has rebooted.

12 Log on to the Ultra M Manager Node.

13 Set the package-c-state-limit CIMC setting.

./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg ospd.cfg --login *<cimc_username> <cimc_user_password>*

14 Verify that the package-c-state-limit CIMC setting has been made.

./ultram_ucs_utils.py --status bios-settings --cfg controller.cfg --login *<cimc_username> <cimc_user_password>*
Look for **PackageCStateLimit** to be set to C0/C1.

15 Update the grub setting with "processor.max_cstate=0 intel_idle.max_cstate=0".

sudo grubby --info=/boot/vmlinuz-`uname -r`
sudo grubby --update-kernel=/boot/vmlinuz-`uname -r` --args="processor.max_cstate=0 intel_idle.max_cstate=0"

16 Verify that the update was successful.

sudo grubby --info=/boot/vmlinuz-`uname -r`
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

17 Reboot the server.

sudo reboot

18 Recheck all CIMC and kernel settings upon reboot.

a Verify CIMC settings

./ultram_ucs_utils.py --status bios-settings --cfg ospd.cfg --login <*cimc_username*> <*cimc_user_password*>

b Verify the processor c-state.

cat /sys/module/intel_idle/parameters/max_cstate
cpupower idle-info

Controlling UCS BIOS Parameters Using *ultram_ucs_utils.py* Script

The ultram_ucs_utils.py script can be used to modify and verify parameters within the UCS server BIOS. This script is in the */opt/cisco/usp/ultram-manager* directory.

Important Refer to the UCS server documentation BIOS documentation for information on parameters and their respective values.

To configure UCS server BIOS parameters:

1 Log on to the Ultra M Manager Node.

2 Modify the desired BIOS parameters.

./ultram_ucs_utils.py --cfg "*config_file_name*" **--login** *cimc_username cimc_user_password* **--mgmt 'set-bios' –-bios-param** *bios_paramname* **–-bios-values** *bios_value1 bios_value2*
Example:

```
./ultram_ucs_utils.py --cfg cmp_17 --login admin abcabc --mgmt 'set-bios --bios-param
biosVfUSBPortsConfig --bios-values vpAllUsbDevices=Disabled vpUsbPortRear=Disabled
```

Example output:

```
2017-10-06 19:48:39,241 - Set BIOS Parameters
2017-10-06 19:48:39,241 - Logging on UCS Server: 192.100.0.25
2017-10-06 19:48:39,243 - No session found, creating one on server: 192.100.0.25
2017-10-06 19:48:40,711 - Login successful to server: 192.100.0.25
2017-10-06 19:48:52,709 - Logging out from the server: 192.100.0.25
2017-10-06 19:48:53,893 - Successfully logged out from the server: 192.100.0.25
```

3 Verify that your settings have been incorporated.

./ultram_ucs_utils.py --cfg "*config_file_name*" **--login** *cimc_username cimc_user_password* **-- status bios-settings**

Example output:

./ultram_ucs_utils.py --cfg cmp_17 --login admin abcabc --status bios-settings

```
2017-10-06 19:49:12,366 - Getting status information from all the servers
2017-10-06 19:49:12,366 - Logging on UCS Server: 192.100.0.25
2017-10-06 19:49:12,370 - No session found, creating one on server: 192.100.0.25
2017-10-06 19:49:13,752 - Login successful to server: 192.100.0.25
2017-10-06 19:49:19,739 - Logging out from the server: 192.100.0.25
2017-10-06 19:49:20,922 - Successfully logged out from the server: 192.100.0.25
2017-10-06 19:49:20,922 -
--------------------------------------------------------------------------
Server IP      | BIOS Settings
--------------------------------------------------------------------------
192.100.0.25   | biosVfHWPMEnable
               |  vpHWPMEnable: Disabled
               | biosVfLegacyUSBSupport
               |  vpLegacyUSBSupport: enabled
               | biosVfPciRomClp
               |  vpPciRomClp: Disabled
               | biosVfSelectMemoryRASConfiguration
               |  vpSelectMemoryRASConfiguration: maximum-performance
               | biosVfExtendedAPIC
               |  vpExtendedAPIC: XAPIC
               | biosVfOSBootWatchdogTimerPolicy
               |  vpOSBootWatchdogTimerPolicy: power-off
               | biosVfCoreMultiProcessing
               |  vpCoreMultiProcessing: all
               | biosVfQPIConfig
               |  vpQPILinkFrequency: auto
               | biosVfOutOfBandMgmtPort
               |  vpOutOfBandMgmtPort: Disabled
               | biosVfVgaPriority
               |  vpVgaPriority: Onboard
               | biosVfMemoryMappedIOAbove4GB
               |  vpMemoryMappedIOAbove4GB: enabled
               | biosVfEnhancedIntelSpeedStepTech
               |  vpEnhancedIntelSpeedStepTech: enabled
               | biosVfCmciEnable
               |  vpCmciEnable: Enabled
               | biosVfAutonumousCstateEnable
               |  vpAutonumousCstateEnable: Disabled
               | biosVfOSBootWatchdogTimer
               |  vpOSBootWatchdogTimer: disabled
               | biosVfAdjacentCacheLinePrefetch
               |  vpAdjacentCacheLinePrefetch: enabled
               | biosVfPCISlotOptionROMEnable
               |  vpSlot1State: Disabled
               |  vpSlot2State: Disabled
               |  vpSlot3State: Disabled
               |  vpSlot4State: Disabled
               |  vpSlot5State: Disabled
               |  vpSlot6State: Disabled
               |  vpSlotMLOMState: Enabled
               |  vpSlotHBAState: Enabled
               |  vpSlotHBALinkSpeed: GEN3
               |  vpSlotN1State: Disabled
               |  vpSlotN2State: Disabled
               |  vpSlotFLOMLinkSpeed: GEN3
               |  vpSlotRiser1Slot1LinkSpeed: GEN3
               |  vpSlotRiser1Slot2LinkSpeed: GEN3
               |  vpSlotRiser1Slot3LinkSpeed: GEN3
               |  vpSlotSSDSlot1LinkSpeed: GEN3
               |  vpSlotSSDSlot2LinkSpeed: GEN3
               |  vpSlotRiser2Slot4LinkSpeed: GEN3
               |  vpSlotRiser2Slot5LinkSpeed: GEN3
               |  vpSlotRiser2Slot6LinkSpeed: GEN3
               | biosVfProcessorC3Report
               |  vpProcessorC3Report: disabled
               | biosVfPCIeSSDHotPlugSupport
               |  vpPCIeSSDHotPlugSupport: Disabled
               | biosVfExecuteDisableBit
               |  vpExecuteDisableBit: enabled
               | biosVfCPUEnergyPerformance
```

```
|  vpCPUEnergyPerformance: balanced-performance
| biosVfAltitude
|  vpAltitude: 300-m
| biosVfSrIov
|  vpSrIov: enabled
| biosVfIntelVTForDirectedIO
|  vpIntelVTDATSSupport: enabled
|  vpIntelVTDCoherencySupport: disabled
|  vpIntelVTDInterruptRemapping: enabled
|  vpIntelVTDPassThroughDMASupport: disabled
|  vpIntelVTForDirectedIO: enabled
| biosVfCPUPerformance
|  vpCPUPerformance: enterprise
| biosVfPchUsb30Mode
|  vpPchUsb30Mode: Disabled
| biosVfTPMSupport
|  vpTPMSupport: enabled
| biosVfIntelHyperThreadingTech
|  vpIntelHyperThreadingTech: disabled
| biosVfIntelTurboBoostTech
|  vpIntelTurboBoostTech: enabled
| biosVfUSBEmulation
|  vpUSBEmul6064: enabled
| biosVfMemoryInterleave
|  vpChannelInterLeave: auto
|  vpRankInterLeave: auto
| biosVfConsoleRedirection
|  vpBaudRate: 115200
|  vpConsoleRedirection: disabled
|  vpFlowControl: none
|  vpTerminalType: vt100
|  vpPuttyKeyPad: ESCN
|  vpRedirectionAfterPOST: Always Enable
| biosVfQpiSnoopMode
|  vpQpiSnoopMode: auto
| biosVfPStateCoordType
|  vpPStateCoordType: HW ALL
| biosVfProcessorC6Report
|  vpProcessorC6Report: enabled
| biosVfPCIOptionROMs
|  vpPCIOptionROMs: Enabled
| biosVfDCUPrefetch
|  vpStreamerPrefetch: enabled
|  vpIPPrefetch: enabled
| biosVfFRB2Enable
|  vpFRB2Enable: enabled
| biosVfLOMPortOptionROM
|  vpLOMPortsAllState: Enabled
|  vpLOMPort0State: Enabled
|  vpLOMPort1State: Enabled
| biosVfPatrolScrub
|  vpPatrolScrub: enabled
| biosVfNUMAOptimized
|  vpNUMAOptimized: enabled
| biosVfCPUPowerManagement
|  vpCPUPowerManagement: performance
| biosVfDemandScrub
|  vpDemandScrub: enabled
| biosVfDirectCacheAccess
|  vpDirectCacheAccess: auto
| biosVfPackageCStateLimit
|  vpPackageCStateLimit: C6 Retention
| biosVfProcessorC1E
|  vpProcessorC1E: enabled
| biosVfUSBPortsConfig
|  vpAllUsbDevices: disabled
|  vpUsbPortRear: disabled
|  vpUsbPortFront: enabled
|  vpUsbPortInternal: enabled
|  vpUsbPortKVM: enabled
|  vpUsbPortVMedia: enabled
| biosVfSataModeSelect
|  vpSataModeSelect: AHCI
```

```
                    | biosVfOSBootWatchdogTimerTimeout
                    |  vpOSBootWatchdogTimerTimeout: 10-minutes
                    | biosVfWorkLoadConfig
                    |  vpWorkLoadConfig: Balanced
                    | biosVfCDNEnable
                    |  vpCDNEnable: Disabled
                    | biosVfIntelVirtualizationTechnology
                    |  vpIntelVirtualizationTechnology: enabled
                    | biosVfHardwarePrefetch
                    |  vpHardwarePrefetch: enabled
                    | biosVfPwrPerfTuning
                    |  vpPwrPerfTuning: os
--------------------------------------------------------------------------------
```