Firmware Management

Firmware Management

Configuring global firmware management enforces a policy in which all domains within a domain group, or maintenance group (v1.5 and above), run a consistent version of the firmware and hardware capability catalog. Enable a global policy resolution when you want to enforce consistent versions of infrastructure firmware among all domains in a domain group, or maintenance group.

Currently, Cisco UCS Central has two different methods for identifying a domain. Prior to v1.4, the firmware update mechanism was based on the domain group assignment on the domain. Now, the update depends on the tag assigned to the domain. If you tag all of the domains with the same Maintenance Group tag, they are upgraded together. If you tag all of the domains in a domain group, and include all sub-domain groups, they are upgraded together.

When a domain group, or maintenance group, consists of multiple Cisco UCS domains, and you have switched to a global firmware management control policy, be careful of any operational (global) firmware policy changes to that group. Cisco UCS Central tries to upgrade all domains that are members of that domain group, or maintenance group, along with any subdomain groups, as defined in the maintenance policy. The default maintenance policy behavior requires user acknowledgment before any updates occur. You can override this behavior by scheduling an acknowledgment or performing an immediate acknowledgment.

Firmware Management for Cisco UCS Domains

Upgrading Cisco UCS domains used to be a manual process, unless you used a Cisco UCS Firmware Upgrade script.

The current release of Cisco UCS Manager includes a firmware automatic-install feature to help automate tasks that were previously manual. Cisco UCS Central builds upon this new feature to help in automating firmware upgrades across multiple Cisco UCS domains.

There are several types of firmware:

- **Infrastructure firmware (A package)**: Refers to the images that run in the I/O modules, the fabric interconnects, and Cisco UCS Manager.
- **Server Blade firmware (B package)**: Refers to the images that run on a physical Cisco UCS blade server BIOS, CIMC, adaptors, and controllers.

Note Currently, Cisco UCS Central does not support direct endpoint upgrades of blades servers. Use a Host Firmware policy and global service profile association to upgrade the blade or server firmware.

- **Rack-Mounted Server firmware (C package)**: Refers to the images that run on a physical Cisco UCS-managed rack-mounted server BIOS, CIMC, adaptors, and controllers.
- **Cisco UCS Mini firmware (E package)**: Refers to the images that run on a physical blade server BIOS, CIMC, adaptors, and controllers that are a part of Cisco UCS Mini.
- **Catalog firmware (T package)**: Refers to capability catalog updates.

The best practice for server firmware is to leverage host firmware packages as part of the service profile definition, to guarantee configuration consistency at the application level.

Recommendations for Firmware Management

There are no implicit maintenance policies for server firmware updates. Therefore, a best practice is to explicitly define a maintenance policy that governs server firmware bundle updates.

Infrastructure firmware updates are disruptive to the server. Therefore, when you create your global service profiles, always enable the user-acknowledgment settings. Using user-acknowledgments in your maintenance policy prevents untimely service disruptions.

You must understand the following firmware management issues.

Service Degradation and Disruption

Any Cisco UCS infrastructure firmware update causes service degradation, because each fabric interconnect must sequentially go through a reboot cycle. To avoid service disruption, ensure that appropriate application-level availability schemes are in place, such as Cisco UCS fabric failover, high-availability (HA), NIC bonding, and host-based storage multipathing.

Pending Acknowledgment

Rebooting fabric interconnects requires explicit acknowledgment from the Cisco UCS Central administrator. For domain/maintenance groups of more than one UCS domain, you can switch the Policy Resolution Control to local for each domain that you do not wish to upgrade. This prevents simultaneous requests to upgrade

UCS domains. Thereafter, changing the Policy Resolution Control to global would cause the upgrade process to proceed.

The default behavior is that you must acknowledge all firmware upgrades (infra-fw) and reboots (fi-reboot) before they proceed. View progress in Cisco UCS Central in the infrastructure firmware upgrade page.

For an individual Cisco UCS domain within Cisco UCS Manager, follow **Operations** > **Firmware** > **FW Operations**.

Infrastructure Firmware Update Process

The update process requires several acknowledgments.

In Cisco UCS Central, multiple updates can proceed in parallel. If you are connected to Cisco UCS Manager, those connections to Cisco UCS Manager are reset during an update. As with standalone Cisco UCS Manager updates, the infrastructure firmware update takes approximately one hour to complete, but can run in parallel across multiple domains. Blade server updates can take longer, depending on the scope of servers involved and whether virtual workloads require migration (vMotion/Life Migration) prior to updating the host.

Firmware management can complement host firmware policies. When loading a new Cisco UCS domain for the first time, use both infrastructure and host firmware automatic installation processes. Ensure that a new domain is current with all low-level host firmware, before releasing it to production.

Host Firmware Packages and Maintenance Policies

Host firmware packages and maintenance policies are both visible and configurable from domain groups and organizations.

The expected behavior for Cisco UCS Central is:

- Global service profiles refer to the host firmware policy that is defined under organization and not the domain group.
- After you acknowledge the maintenance policy, Cisco UCS Central pulls the host firmware package, maintenance policy, and any other referenced policies, from Cisco UCS Manager to Cisco UCS Central.
- A local service profile can either reference a local host firmware policy and a local maintenance policy, a global host firmware policy, or global maintenance policy.

The best practice is to configure and use host firmware packages, maintenance policies, and schedules exclusively from the organization.

Tags

Cisco UCS Central uses tags to allow users to group objects outside of the Organization or Domain Group structures. You can add tags to various objects:

- Policies
- Logical resources
- Physical inventory components

You can use the following types of tags:

- **System-defined tags**:
  - Maintenance Group tags—Used for Infrastructure Firmware updates
  - Operating System for HCR tags—Used for Hardware Compatibility Reports
  - Adapter Driver for HCR tag—Used for Hardware Compatibility Reports

Note When applying a maintenance group tag to a domain for Infrastructure Firmware updates, you can only apply one maintenance group tag to a domain. For Operating System for HCR tags, you can also only apply one tag to a server. However, you can add multiple Adapter driver for HCR tags to adapters, if needed.

- **User-defined tags—**Tags that are created by users and have specific, user-designated values. They display in the Tag Management page, under the **Tag Types** tab.

  All of the tags added to objects based on the user-defined tag types display in the Tags tab on the Tag Management page. They also display in the Add tags drop-down list. Create these tags to manage your domains and organizations.

- **Basic tags—**Free text tags that can allow any value.

Tagging Domains with an API Script

The best method for tagging domains is to apply tags using the Cisco XML API script. The `configConfMo` method configures the specified managed object in a single subtree. Following is a sample script for tagging domains to include them in maintenance groups for infrastructure firmware updates:

```
<configConfMo
    dn="holder/tag-ep"
    cookie="<real_cookie>"
    inHierarchical="false">
    <inConfig>
       <tagInstance
          defName="Maintenance Group"
          taggedObjectDn="compute/sys-1009"
          value="tagTest"
          status="created"
          />
    </inConfig>
</configConfMo>
```

- The <*cookie*> variable is for an actual cookie. When you log in to a session using the API script, the request returns a cookie for that session. Paste that 47-character string into your script for the cookie.
- The <*defname*> variable specifies the tag type. It is a required field. In the example above, the <*defname*> is maintenance group, so these tags are used to tag domains for inclusion in maintenance groups for infrastructure firmware updates.
- The <*taggedObjectDN*> variable defines the object to be tagged. In the example above, the *taggedObjectDn* is the distinguished name of the domain to be tagged. It indicates that we are adding a Maintenance Group tag to the domain with a distinguidhed name of 'compute/sys-1009'

- The <*value*> variable is the name or value of the tag.

Refer to the Cisco UCS Central XML API Programmer's Guide for more information.

Hardware Compatibility List

The Hardware compatibility list (HCL) tool allows users to verify that all third party drivers, hardware, etc. are compatible with the Cisco UCS Central system. Before running a hardware compatibility report, you must:

- Associate relevant servers with global service profiles because an HCL only runs on servers associated with a global service profile.
- Apply Operating System for HCR and Adapter Driver for HCR tags to your servers and adapters. The best method for this is to use an API script.
- Download the latest version of the hardware compatibility list from Cisco.com. Obtain hardware compatibility lists by scheduling periodic hardware compatibility list synchronization, by synchronizing the list on demand, or by downloading the list externally and importing it.

The HCL report shows if an operating system or an adapter is incompatible with Cisco UCS Central.

Note If Cisco UCS Central detects an incompatible operating system, it stops generating.

Click **View Compatible Operating systems** to view a list of operating systems that are compatible with your current setup. After you change the operating system, run the report again and click the Refresh icon to view the new status.

Another method for running the HCL report is through a Host Firmware Package Policy. See the Cisco UCS Central Server Management Guide, Release 1.5 for more information.

HCL OS and Adapter Tags

Following are sample scripts for adding OS tags and Adapter tags.

```
<configConfMo
dn="holder/tag-ep"
cookie="<real_cookie>"
inHierarchical="false">
  <inConfig>
    <tagSoftwareInst
    defName="Operating System for HCR"
    taggedObjectDn="compute/sys1/ch1/bl3"
    value=""
    vendor="CentOS"
    version="CentOS 6.6"
    status="created"
    />
    </inConfig>
</configConfMo>

<configConfMo
dn="holder/tag-ep"
cookie="<real_cookie>"
inHierarchical="false">
  <inConfig>
    <tagDriver
       defName='Adapter Driver for HCR'
```

```
        taggedObjectDn='compute/sys1/ch1/bl4/ad1'
        protocol='Ethernet'
        vendor='Cisco'
        version='2.3.0.20'
        value=''
        status='created'
       />
    </inConfig>
</configConfMo>
```

- The <*cookie*> variable is for an actual cookie. When you log in to a session using the API script, the request returns a cookie for that session. Paste that 47-character string into your script for the cookie.
- The <*defname*> variable specifies the tag type. This is a required field. In the example above, the <*defname*> is Operating System for HCR in the first example and Adapter Driver for HCR in the second. These tags are used to tag servers and adapters for inclusion in the HCL report.
- The <*taggedObjectDN*> variable defines the name or ID for the device. This field is required.
- The <*vendor*> and <*version*> variables reflect the name of the vendor and software version. These are required fields.

Note If you need to verify different driver types, create multiple `tagDriver` tags.

Refer to the Cisco UCS Central XML API Programmer's Guide for more information.