API Backwards Compatibility

API Backwards Compatibility

Backwards Compatibility Overview

The Cisco Unified Communications Domain Manager API is versioned and while the latest API and models are in use, the system is backwards compatible with earlier API versions for several models and operations on these. This means that API requests follow the schema and data as specified by the API version.

From 10.6.2 onward, the default web server security protocol has been set to TLSv1.2 and that TSLv1 is no longer supported. SSLv3 can be enabled from the command line with the command **web sslv3 <on/off>**

It is therefore easier to use a network device or custom code written for an earlier Cisco Unified Communications Domain Manager version. An HTTP Request parameter or Header change in developer code may be required for API calls.

Third-party clients that are written to use a particular API version continues to work against newer servers as long as the Cisco Unified Communications Domain Manager server continues to support the API version used by the client. Cisco Unified Communications Domain Manager supports APIs from the current release and two previous releases (N-2).

For device models, an internal version mapping table for API versions and device versions is also maintained. This table creates a fixed mapping of device versions to API version. The mapping is created on a principle of the latest supported device version at the time of the API version release. For example, the version 10.0 UC Application schema is mapped to API version 10.1.2.

No transforms are carried out on Relation models, because transforms are carried out on their component Data- and Device models.

Tools such as Bulk Load and Search, are backwards compatible from an operations and URL structure point of view. However, the data processed or generated by the tools do not support Backwards Compatibility. Data

must be adapted to conform to the latest schema when presenting it to the API. Data obtained from the Tool interfaces are interpreted according to the latest schema.

Note For Update operations: if a model schema was changed so that a new field is added to a schema within a list of objects and data exists in the field for a current instance on the system, then the data in the new field cannot be maintained when updating the instance through a backwards compatible request. Request data replaces all the data within a list of objects. There is currently no workaround for this issue.

API Version

The API Version is represented in major.minor.revision format, for example: 10.1.2, 10.6.1, 10.6.2, or 10.6.3.

The API Version can be seen in the meta section of the resource as follows:

```
"meta": {
"tags": [],
"pkid": "",
"schema_version": "0.1",
"hierarchy": "sys",
"version_tag": "0.2",
"api_version": "10.6.1",
"model_type": "data/DataModel"
},
```

Supported Models and Methods

Supported models are:

- Data Models
- Device Models
- Relations
- Views

The supported HTTP methods on models from the API are:

- GET
- POST
- PUT

Versions Supported

Support for backwards compatibility was introduced in 10.6(1) release.

- The 10.6(2) release is backwards compatible with the 10.6(1) release.
- The 10.6(1) release is backwards compatible with the 10.1(2) release.
- The 10.6(1) release is **not** compatible with the 10.1(1) release.
- The 10.1(2) release is **not** compatible with the 10.1(1) release.

Specifying the API Version

Third-party API clients **must** specify the api_version when integrating with Cisco Unified Communications Domain Manager 10.6(x). This is required if the third-party client wants to use backwards compatibility, since it ensures that the client continues to receive consistent schemas. Specifying the API version can be done in one of two ways.

In the Query Parameter

```
GET http://localhost/api/data/Countries/?hierarchy=[hierarchy]
&schema=true&format=json&api_version=10.6.1
```

The Query Parameter approach is the recommended method for a client to specify the API Version.

In the Request Header

```
GET http://localhost/api/data/Countries/?hierarchy=[hierarchy]
&schema=true&format=json

Request headers
X-Version: 10.6.1
```

Omitting the API Version

If the API Version is omitted, then the following behavior is expected:

- If the URL contains /v0/, the 10.1(2) API schemas are used.
- If the URL does not contain /v0/, the most recent version API schemas are used.

Backwards Compatibility Exceptions

10.6(3) Exceptions

Prior to 10.6(3), the SyncTo hierarchy of the user in the Provisioning Status was set to the hierarchy where the User is created. Whenever the user is moved up or down the SyncTo was updated to the hierarchy where it is moved. To move subscribers between Sites, the SyncTo hierarchy behavior has been changed in a backwards incompatible way. The SyncTo Hierarchy focuses on updating the syncTo in ProvisionalStatusDAT when the user changes from a “Manual” user to a “Subscriber” user pushed to a Cisco Unified Communications Manager. In 10.6(3) the strategy is changed as follows:

- Manually created Users (User Management not Subscriber Management) have the SyncTo hierarchy set to the hierarchy the user is created.
- Manually created Users (User Management) that are pushed to Cisco Unified Communications Manager (becoming a Subscriber) from Customer or Site has their SyncTo hierarchy updated to that of the Cisco Unified Communications Manager.

Note If the user SyncTo hierarchy is above the Cisco Unified Communications Manager hierarchy when it was created, then the SyncTo does not change and remains at the upper hierarchy.

- Manually created Subscribers (Subscriber Management) have there SyncTo hierarchies set to to that of the Cisco Unified Communications Manager.

- LDAP users are not affected (do not have their SyncTo hierarchy changed).
- If a user is created on an intermediate node or site directly (either manually or through QuickAddSubscriber) and then pushed to Cisco Unified Communications Manager, the user is scoped only to the hierarchy where he or she was created.

10.6(1) Exceptions

The following changes in Cisco Unified Communications Domain Manager may impact API backwards compatibility with the previous version:

- Customer Management – Customer name change is now allowed.
- Unicode – Customer names may not have spaces.
- Localization – Language codes are now four characters.

10.1(2) Exceptions

The 10.1(2) release is not backwards compatible with the 10.1(1) release.

API Version Differences

There are some differences in customer facing models in this release with respect to the previous releases. These differences are captured in the following document.

http://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/hcs/10_6_2/CUCDM_10_6_2/API_Reference_Docs/CUCDM_API_Schema_Diffs_From1061HCM_Standard_To1062HCM_Standard.pdf

API Backward Compatibility and Import

The Cisco Unified Communications Domain Manager 10.6(x) system maintains a Data Model version data store containing all versions that have been imported onto the system.

While there is always a current version of a Data Model in use on the system, a check is carried out during the import of data:

- If the current version is newer than the definition of the imported data, then the imported definition data is flagged internally as `automigration: false` to prevent resources from auto-migrating from a newer version to an older version.
- Importing an older version will not replace the latest definition as the default schema. The older version will only be added to the version store.

The snippet example below shows the `automigration` attribute:

```
{
   "meta": {},
   "resources": [
      {
         "data": {
            "name": "test_mig_dm"
             ...
         },
         "meta": {
            "hierarchy": "sys",
```

```
            "model_type": "data/DataModel",
            "schema_version": "0.1",
            "version_tag": "0.3",
            "automigration": false
        }
    }
]
}
```

This model definition version store makes it possible for version definition imports to be sequence independent, allowing a freshly installed system to construct the version history for backwards compatibility.

HIL API Backward Compatibility

HIL currently supports API versions for v10_1_2 and v10_6_1. You can use the following commands to get the current API version of the target system and to set the API version of the target system:

- `show hcs hil target apiversion` - To get the current API version of target system
- `set hcs hil target apiversion` - To set the API version of target system

Note If there are any active HIL sessions, CLI will prompt the number of active sessions and will check for confirmation before changing the API version