



API Backwards Compatibility

- [API Backwards Compatibility, page 1](#)
- [Backwards Compatibility Exceptions, page 3](#)
- [API Version Differences, page 3](#)
- [HIL API Backward Compatibility, page 3](#)

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 a number of models and operations on these. This means that API requests follow the schema and data as specified by the API version.

Third party clients that are written to use a particular API version should continue 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).

API Version

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

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(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(1). This is required if the third party client wants to make use of backwards compatibility, since it ensures 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

10.1(2)

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

10.6(1)

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 4 characters.

API Version Differences

There are some differences in customer facing models in release 10.6.(1) with respect to the previous release, 10.1(2). These differences are captured in the following document.

http://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/hcs/10_6_1/CUCDM_10_6_1/API_Guide/CUCDM_API_Schema_Diffs_HCM_Standard.pdf

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
