



Cisco Unity Connection Provisioning Interface (CUPI) API for User Account Settings

- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Alternate Names, on page 1](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Changing Passwords and Password Settings, on page 4](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Changing Primary and Alternate Extensions, on page 5](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Class of Service \(COS\), on page 7](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Enabling Single Inbox Example, on page 10](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Notification Devices, on page 11](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Private List Members , on page 24](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Private Lists, on page 27](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- SMTP Proxy Addresses , on page 29](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- User Import, on page 31](#)

Cisco Unity Connection Provisioning Interface (CUPI) API -- Alternate Names

About Alternate Names

This page contains information on how to use the API to create, list, update, and delete Alternate Names. Alternate Names are shared by users, contacts, distribution lists, private lists, and VPIM locations. Each of these objects can have Alternate Names associated with them.

Listing and Viewing

The following is an example of a GET that lists all Alternate Names:

```
GET http://<connection-server>/vmrest/alternatenames
```

The following is the response from the above GET request:

```

200
OK
<?xml version="1.0" encoding="UTF-8"?>
<AlternateNames total="5">
  <AlternateName>
    <URI>/vmrest/alternatenames/a2ec7ba8-db89-4135-8cde-2fce23e59616</URI>
    <FirstName>Ooser</FirstName>
    <LastName>Aye</LastName>
    <ObjectId>a2ec7ba8-db89-4135-8cde-2fce23e59616</ObjectId>
    <GlobalUserObjectId>bde24d71-95fa-4ba8-bf1b-0e19a4e9a68b</GlobalUserObjectId>
    <GlobalUserURI>/vmrest/globalusers/bde24d71-95fa-4ba8-bf1b-0e19a4e9a68b</GlobalUserURI>
  </AlternateName>
  <AlternateName>
    <URI>/vmrest/alternatenames/c10257ad-886a-451d-9d96-09499dbf5cf1</URI>
    <FirstName>allvoicemaleusers</FirstName>
    <ObjectId>c10257ad-886a-451d-9d96-09499dbf5cf1</ObjectId>
    <DistributionListObjectId>a9a648f7-bf32-4851-a0c5-62c0165116ae</DistributionListObjectId>
  </AlternateName>
  <DistributionListURI>/vmrest/distributionlists/a9a648f7-bf32-4851-a0c5-62c0165116ae</DistributionListURI>
  </AlternateName>
  <AlternateName>
    <URI>/vmrest/alternatenames/2b5b5a9a-1721-4ca6-a1e6-92d375f6c761</URI>
    <FirstName>khan</FirstName>
    <LastName>ah</LastName>
    <ContactObjectId>95beb3f7-b142-4ca5-861f-e36d65aef463</ContactObjectId>
    <ContactURI>/vmrest/contacts/95beb3f7-b142-4ca5-861f-e36d65aef463</ContactURI>
    <ObjectId>2b5b5a9a-1721-4ca6-a1e6-92d375f6c761</ObjectId>
  </AlternateName>
  <AlternateName>
    <URI>/vmrest/alternatenames/d6d06ae0-6c63-4944-9426-ce4027aa67fd</URI>
    <FirstName>privut lyst</FirstName>
    <ObjectId>d6d06ae0-6c63-4944-9426-ce4027aa67fd</ObjectId>
    <PersonalGroupObjectId>10966c03-e64b-4c3d-9809-990b62865c6a</PersonalGroupObjectId>
  </AlternateName>
  <AlternateName>
    <URI>/vmrest/alternatenames/44e03050-5e7e-4792-be69-96e1e7511224</URI>
    <FirstName>vpim lokaychion aye</FirstName>
    <ObjectId>44e03050-5e7e-4792-be69-96e1e7511224</ObjectId>
    <LocationObjectId>6aacd387-72e9-4971-82e5-b04a7a9790ad</LocationObjectId>
  </AlternateName>
  <LocationURI>/vmrest/locations/connectionlocations/6aacd387-72e9-4971-82e5-b04a7a9790ad</LocationURI>
  </AlternateName>
</AlternateNames>

```

In this example, the first Alternate Name in the list belongs to a user as denoted by the `GlobalUserObjectId` field. The second Alternate Name belongs to a Distribution List, and so on.

For Distribution Lists, Private Lists, and VPIM Locations, only the `FirstName` field is used to provide the Alternate Name. These objects only have a single Display Name, they do not have separate first and last names.

To retrieve a sorted list of all Alternate Names, add the following query parameter:

```
sort=(column [asc | desc])
```

For example, to retrieve a list of all Alternate Names sorted by Object Id in ascending order:

```
GET http://<connection-server>/vmrest/alternatenames?sort=(objectid%20asc)
```

To retrieve a specific Alternate Name by its object ID:

```
GET http://<connection-server>/vmrest/alternatenames/<objectid>
```

Searching

To retrieve a list of Alternate Names that meet a specified search criteria, add the following query parameter to a GET:

```
query=(column [is | startswith] value)
```

For example, to find all Alternate Names with a GlobalUserObjectId that equals "bde24d71-95fa-4ba8-bf1b-0e19a4e9a68b" (this is a search to find all Alternate Names for a specific User):

```
GET
http://<connection-server>/vmrest/alternatenames?query=(globaluserobjectid%20is%20bde24d71-95fa-4ba8-bf1b-0e19a4e9a68b)
```

To find Alternate Names for various objects, use the following object id columns:

- For Users, use GlobalUserObjectId.
- For Contacts, use ContactObjectId.
- For Distribution Lists, use DistributionListObjectId.
- For Private Lists, use PersonalGroupObjectId.
- For VPIM Locations, use LocationObjectId.

Creating

The required fields for creating an Alternate Name are a parent object id, a First Name, and for some objects a Last Name. To determine what parent object id column to use to create the new Alternate Name, use the list found in the Searching section above. For example, if you are creating an Alternate Name for a User, then a GlobalUserObjectId needs to be provided.

The following is an example of a POST that creates an Alternate Name for a User with a First Name of "Mike" and a Last Name of "Wholebert" whose Global User Object Id is "bde24d71-95fa-4ba8-bf1b-0e19a4e9a68b":

```
POST http://<connection-server>/vmrest/alternatenames

<?xml version="1.0" encoding="UTF-8"?>
  <AlternateName>
    <FirstName>Mike</FirstName>
    <LastName>Wholebert</LastName>
    <GlobalUserObjectId>bde24d71-95fa-4ba8-bf1b-0e19a4e9a68b</GlobalUserObjectId>
  </AlternateName>
```

When creating Distribution Lists, Private Lists, or VPIM Locations, only the FirstName field is used to provide the Alternate Name

Updating

The following is an example of a PUT request that modifies the First Name and Last Name of an existing Alternate Name:

```
PUT http://<connection-server>/vmrest/alternatenames/<objectid>

<?xml version="1.0" encoding="UTF-8"?>
  <AlternateName>
    <FirstName>Mick</FirstName>
    <LastName>Holebert</LastName>
  </AlternateName>
```

Deleting

The following is an example of a DELETE request that deletes an Alternate Name:

```
DELETE http://<connection-server>/vmrest/alternatenames/<objectid>
```

Cisco Unity Connection Provisioning Interface (CUPI) API -- Changing Passwords and Password Settings

Changing a User's PIN

A user's PIN is the password that a user must enter over the phone to sign in to their mailbox, so that they can listen to or send new messages by phone. To change a user's PIN you need the object ID of the user. The following example changes the user's PIN to 635241.

```
PUT vmrest/users/<objectid>/credential/pin

<?xml version="1.0" encoding="UTF-8"?>
<Credential>
  <Credentials>635241</Credentials>
</Credential>
```

Changing a User's Password

A user's password is the web application password required to interact with web applications, such as the PCA or the Inbox applications. The following example changes the user's password to My1stPassword.

```
PUT vmrest/users/<objectid>/credential/password

<?xml version="1.0" encoding="UTF-8"?>
<Credential>
  <Credentials>My1stPassword</Credentials>
</Credential>
```

Setting the *User Cannot Change Password* Flag

To configure a PIN or password so that the user cannot change it, use the CantChange element.

Here is an example of changing a user's PIN such that the user cannot change it.

```
PUT vmrest/users/<objectid>/credential/pin

<?xml version="1.0" encoding="UTF-8"?>
<Credential>
  <CantChange>true</CantChange>
</Credential>
```

Here is an example of changing a user's password such that the user cannot change it.

```
PUT vmrest/users/<objectid>/credential/password

<?xml version="1.0" encoding="UTF-8"?>
<Credential>
  <CantChange>true</CantChange>
</Credential>
```

Note that the only difference in these examples is the URI; they both use the same Credentials object.

Cisco Unity Connection Provisioning Interface (CUPI) API -- Changing Primary and Alternate Extensions

About Changing Primary and Alternate Extensions

A common scenario is having a primary extension that needs to be changed, but the old primary extension needs to be kept around as an alternate extension for that same user. This document describes the steps to make such a change. It shows how to either create a new alternate extension for a user, or update an existing alternate extension.

Viewing the Primary Extension

The user's primary extension is the DTMFAccessID field on the user object. The following is an example of a GET that lists the user object:

```
GET /vmrest/users/<userobjectid>
```

The user's primary extension will also show up in their list of alternate extensions with an IdIndex of 0.

Updating the Primary Extension

The following is an example of a PUT that modifies the DTMFAccessID of a user to 2001:

```
PUT /vmrest/users/<userobjectid>

<?xml version="1.0" encoding="UTF-8"?>
<User>
  <DtmfAccessId>2001</DtmfAccessId>
</User>
```

The user's primary extension can also be updated through the user's alternate extensions URI by executing a PUT that modifies the alternate extension with an IdIndex of 0.

Creating a New Alternate Extension

The following is an example of a POST that creates an alternate extension of 2000 for a user. The IdIndex field is explained in further detail below.

```
POST /vmrest/users/<userobjectid>/alternateextensions

<?xml version="1.0" encoding="UTF-8"?>
<AlternateExtension>
  <IdIndex>1</IdIndex>
  <DtmfAccessId>2000</DtmfAccessId>
</AlternateExtension>
```

Updating an Existing Alternate Extension

The following is an example of a PUT that modifies the DTMFAccessID of an alternate extension on a user to 2000. The IdIndex field is explained in further detail below.

```
PUT /vmrest/users/<userobjectid>/alternateextensions/<alternateextensionobjectid>

<?xml version="1.0" encoding="UTF-8"?>
<AlternateExtension>
  <DtmfAccessId>2000</DtmfAccessId>
</AlternateExtension>
```

Explaining the IdIndex Field

The IdIndex field indicates what type of phone this extension represents. It does not affect the behavior of the alternate extension, it simply is a way to categorize and order the alternate extensions.

Cisco Unity Connection allows a user to have 9 administrator-defined alternate extensions and 10 user-defined alternate extensions (if allowed by COS). Only one alternate extension is associated with each IdIndex, and every extension must have an IdIndex.

Non-alternate extensions (primary, distribution list, location, etc.) have an IdIndex containing a value of 0. Administrator-defined alternate extensions use the range of 1-9. User-defined alternate extensions use the range of 11-20. User-defined alternate extensions cannot be created by administrators, only by users (if allowed by their COS).

In Cisco Unity Connection Administration, administrator-defined alternate extensions have their IdIndex translated into a **Phone Type**, which is simply a text tag to help you categorize your alternate extensions. The table below shows the translations.

- 1 = Work Phone
- 2 = Work Phone 2
- 3 = Home Phone
- 4 = Home Phone 2
- 5 = Mobile Phone
- 6 = Mobile Phone 2
- 7 = Work Fax

8 = Work Fax 2

9 = Home Fax

Cisco Unity Connection Provisioning Interface (CUPI) API -- Class of Service (COS)

About Classes of Service (COSes)

This page contains information on how to use the API to create, list, update, and delete classes of service.

Listing and Viewing

The following is an example of a GET request that lists all Classes of Service:

```
GET http://<connection-server>/vmrest/coses
```

The following is the response from the above GET request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Coses total="2">
  <Cos>
    <URI>/vmrest/coses/e0da866c-07d4-41ae-8ca9-e334e4732ea7</URI>
    <ObjectId>e0da866c-07d4-41ae-8ca9-e334e4732ea7</ObjectId>
    <DisplayName>Voice Mail User COS</DisplayName>
  </Cos>
  <Cos>
    <URI>/vmrest/coses/38a57ca9-406d-4dec-9fc0-d7fc54ef5dff</URI>
    <ObjectId>38a57ca9-406d-4dec-9fc0-d7fc54ef5dff</ObjectId>
    <DisplayName>System</DisplayName>
  </Cos>
</Coses>
```

To retrieve a sorted list of all Classes of Service, add the following query parameter: sort=(column [asc|desc])

For example, to retrieve a list of all Classes of Service sorted by their display names in ascending order:

```
GET http://<connection-server>/vmrest/coses?sort=(displayname%20asc)
```

To retrieve a specific Class of Service by its object ID:

```
GET http://<connection-server>/vmrest/coses/<objectid>
```

The following is the response from the above GET request (requesting a single Class of Service by object ID) done on the factory default Voice Mail User Cos:

```

<?xml version="1.0" encoding="UTF-8"?>
<Cos>
  <URI>/vmrest/coses/a9a3a677-31f1-4edb-acd0-6a2135e15995</URI>
  <ObjectId>a9a3a677-31f1-4edb-acd0-6a2135e15995</ObjectId>
  <AccessFaxMail>>false</AccessFaxMail>
  <AccessTts>>false</AccessTts>
  <CallHoldAvailable>>false</CallHoldAvailable>
  <CallScreenAvailable>>false</CallScreenAvailable>
  <CanRecordName>>true</CanRecordName>
  <FaxRestrictionObjectId>d762807b-fbab-4b46-b21b-bf9e37648d0a</FaxRestrictionObjectId>
  <ListInDirectoryStatus>>true</ListInDirectoryStatus>
  <LocationObjectId>7fe7907a-2222-482e-83cb-d8d4ad32adc1</LocationObjectId>

<LocationURI>/vmrest/locations/connectionlocations/7fe7907a-2222-482e-83cb-d8d4ad32adc1</LocationURI>

  <MaxGreetingLength>90</MaxGreetingLength>
  <MaxMsgLength>300</MaxMsgLength>
  <MaxNameLength>30</MaxNameLength>
  <MaxPrivateDlists>25</MaxPrivateDlists>
  <MovetoDeleteFolder>>true</MovetoDeleteFolder>

<OutcallRestrictionObjectId>1c7c3ebf-391e-44f6-8ee2-3717049c614d</OutcallRestrictionObjectId>

  <PersonalAdministrator>>true</PersonalAdministrator>
  <DisplayName>Voice Mail User COS</DisplayName>
  <XferRestrictionObjectId>54caa5d6-0ec6-49c9-a80f-52ae743c256f</XferRestrictionObjectId>
  <Undeletable>>true</Undeletable>
  <WarnIntervalMsgEnd>0</WarnIntervalMsgEnd>
  <CanSendToPublicDl>>true</CanSendToPublicDl>
  <EnableEnhancedSecurity>>false</EnableEnhancedSecurity>
  <AccessVmi>>false</AccessVmi>
  <AccessLiveReply>>false</AccessLiveReply>
  <UaAlternateExtensionAccess>0</UaAlternateExtensionAccess>
  <AccessCallRoutingRules>>false</AccessCallRoutingRules>
  <WarnMinMsgLength>0</WarnMinMsgLength>
  <SendBroadcastMessage>>false</SendBroadcastMessage>
  <UpdateBroadcastMessage>>false</UpdateBroadcastMessage>
  <AccessVui>>false</AccessVui>
  <ImapCanFetchMessageBody>>true</ImapCanFetchMessageBody>
  <ImapCanFetchPrivateMessageBody>>false</ImapCanFetchPrivateMessageBody>
  <MaxMembersPVL>99</MaxMembersPVL>
  <AccessIMAP>>false</AccessIMAP>
  <ReadOnly>>false</ReadOnly>
  <AccessAdvancedUserFeatures>>false</AccessAdvancedUserFeatures>
  <AccessUnifiedClient>>false</AccessUnifiedClient>
  <RequireSecureMessages>4</RequireSecureMessages>
  <AccessOutsideLiveReply>>false</AccessOutsideLiveReply>
  <AccessSTT>>false</AccessSTT>
  <EnableSTTSecureMessage>0</EnableSTTSecureMessage>
</Cos>

```

RequireSecureMessages is a custom type that controls how secure messages are generated by users. It can be set to mark all messages as secure (1), never mark messages as secure (2), ask the user if they want to mark a message as secure (3), or only mark private messages as secure (4).

One can also use CUPI For End Users to retrieve one's own Class of Service with the following request:

```
GET /vmrest/user/cos
```


Searching

To retrieve a list of Classes of Service that meet a specified search criteria, add the following query parameter: query=(column [is|startswith] value)

For example, to find all Classes of Service with a display name that starts with "System":

```
GET http://<connection-server>/vmrest/coses?query=(displayname%20startswith%20System)
```

Creating

The only required field for creating a COS is DisplayName. All other fields are optional.

The following is an example of a POST request that creates a COS with the display name "My New COS":

```
POST http://<connection-server>/vmrest/coses

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Cos>
  <DisplayName>My New COS</DisplayName>
</Cos>
```

The following is the response from the above POST request:

```
201
Created
/vmrest/coses/8ae8b4a2-a25b-4f16-95d9-77abcb91b764
```

Updating

The following is an example of a PUT request that modifies the display name of an existing COS:

```
PUT http://<connection-server>/vmrest/coses/<objectid>

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Cos>
  <DisplayName>My New CoS</DisplayName>
</Cos>
```

The following is the response from the above PUT request:

```
204
No Content
null
```

Deleting

The following is an example of a DELETE request that deletes a Class of Service:

```
DELETE http://vmrest/coses/<objectid>
```

The following is the response from the above DELETE request:

```
204
No Content
null
```

Cisco Unity Connection Provisioning Interface (CUPI) API -- Enabling Single Inbox Example

Enabling Single Inbox for a User

This document explains how to use the API to enable the Single Inbox feature for a user. The API being used in all these examples is CUPI.



Note The Single Inbox feature is available only in Cisco Unity Connection 8.5 and later.

Update Class of Service

To allow a user to use the Single Inbox feature, the Class of Service must allow it. The **AccessIMAP** field controls access to the Single Inbox feature. This field must be set to "true."

Here is an example of modifying an existing Class of Service to allow access to the Single Inbox feature:

```
PUT /vmrest/coses/<cosobjectid>
<?xml version="1.0" encoding="UTF-8"?>
<Cos>
  <AccessIMAP>true</AccessIMAP>
</Cos>
```

Create or Update Unified Messaging Service

Unified Messaging Services cannot currently be created or updated through the API. Any configuration changes will need to be done through another interface such as Cisco Unity Connection Administration.

Create or Update Unified Messaging Accounts

To enable the Single Inbox feature for a user, the user must have a correctly configured Unified Messaging Account in which the **EnableMailboxSynCapability** field is set to "true." Note that prior to Connection 8.5, Unified Messaging Accounts were referred to as External Service Accounts, and this is how the API refers to them.

Here is an example of modifying an existing Unified Messaging Account to enable the Single Inbox feature:

```
PUT /vmrest/users/<userobjectid>/externalserviceaccounts/<externalserviceaccountobjectid>
<?xml version="1.0" encoding="UTF-8"?>
<ExternalServiceAccount>
  <EnableMailboxSynchCapability>true</EnableMailboxSynchCapability>
</ExternalServiceAccount>
```

Cisco Unity Connection Provisioning Interface (CUPI) API -- Notification Devices

About Notification Devices

This page contains information on how to use the API to create, list, update, and delete Notification Devices.

Cisco Unity Connection supports four different types of Notification Devices:

- Phone Devices
- Pager Devices
- SMS Devices
- SMTP Devices

The generic Notification Device resource contains the data fields that are common to most or all of the specialized Device resources, which in turn may contain additional data fields that are specific to certain Device resources. A listing of all data fields, including a description of what they mean and which specialized Device types support them, can be found later in this document.

An API user can retrieve a complete list of all Notification Devices or a specific Notification Device, but this is a read-only resource. In order to create, update, or delete a Notification Device, an API user must use the appropriate specialized Device resource. In other words, an API user can retrieve all of a user's Notification Devices, but if they then want to update a particular Phone Device, they must do the update operation on the Phone Device resource, not the Notification Device resource. This will be explained in more detail below.

The Notification Device resource and the four specialized Device resources contain data fields from the Device objects in the database, plus some data fields from the Notification Rule object. From an API user's perspective, this detail may not usually be important, but it is worth mentioning that the Device resources collapse two database objects into a single resource. This can be safely done because there is always a one-to-one mapping in the database of Notification Devices and Notification Rules.

The factory default User Template for Voicemail Users contains five Notification Devices: three Phone Devices (Home, Work, and Mobile), one Pager Device, and one SMTP Device. Any of these default Notification Devices can be modified, and new Devices of any of four specialized types can be created or deleted. Note that Notification Devices for User Templates are not currently exposed via the API, only Notification Devices for Users are exposed.

Listing and Viewing

An API user can list all generic Notification Device resources or all Device resources of a specified type (Phone, Pager, SMS, or SMTP).

Listing and Viewing Notification Devices

The following is an example of a GET that lists all Notification Devices of all types for the specified User:

```

GET https://<connection-server>/vmrest/users/<UserObjectId>/notificationdevices
{noformat}
The following is the response from the above *GET* request. Notice that this list contains
a mix of different types of specialized Devices, since the request was for a list of all
Notification Devices. The Type field indicates the specialized Device type for each
Notification Device (1 = Phone, 2 = Pager, 4 = SMTP, 5 = SMS).
{noformat}
200
OK

<?xml version="1.0" encoding="UTF-8"?>
<NotificationDevices total="5">
  <NotificationDevice>

<URI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa/notificationdevices/2b13dda8-6249-45b2-9a87-abalb27alf95</URI>

  <SubscriberObjectId>70c5d764-b2f3-498a-b16e-1a4d2a369dfa</SubscriberObjectId>
  <UserURI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa</UserURI>
  <ObjectId>2b13dda8-6249-45b2-9a87-abalb27alf95</ObjectId>
  <DisplayName>Pager</DisplayName>
  <Active>>false</Active>
  <BusyRetryInterval>5</BusyRetryInterval>
  <Type>2</Type>
  <DialDelay>1</DialDelay>
  <MaxBody>512</MaxBody>
  <MaxSubject>64</MaxSubject>
  <RetriesOnBusy>4</RetriesOnBusy>
  <RetriesOnRna>4</RetriesOnRna>
  <RingsToWait>4</RingsToWait>
  <RnaRetryInterval>15</RnaRetryInterval>
  <SendCount>>true</SendCount>
  <WaitConnect>>true</WaitConnect>
  <MediaSwitchObjectId>6ecaa89b-2d29-4a21-8013-75c154ee58f5</MediaSwitchObjectId>

<PhoneSystemURI>/vmrest/phonesystems/6ecaa89b-2d29-4a21-8013-75c154ee58f5</PhoneSystemURI>

  <TransmitForcedAuthorizationCode>>false</TransmitForcedAuthorizationCode>
  <DeviceName>Pager</DeviceName>
  <PromptForId>>false</PromptForId>
  <SendCallerId>>true</SendCallerId>
  <SendPcaLink>>false</SendPcaLink>
  <Undeletable>>true</Undeletable>
  <MediaSwitchDisplayName>PhoneSystem</MediaSwitchDisplayName>
  <DetectTransferLoop>>false</DetectTransferLoop>
  <SuccessRetryInterval>1</SuccessRetryInterval>
  <RetriesOnSuccess>0</RetriesOnSuccess>
  <EventList>NewVoiceMail</EventList>
  <ScheduleSetObjectId>d7ea64ca-e3fc-4d0a-8bd5-ef2337a31b97</ScheduleSetObjectId>
  <InitialDelay>0</InitialDelay>
  <RepeatInterval>0</RepeatInterval>
  <RepeatNotify>>false</RepeatNotify>
  </NotificationDevice>
</NotificationDevice>

<URI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa/notificationdevices/0f4ce029-1939-4ae0-9dee-9d84c77fadae</URI>

  <SubscriberObjectId>70c5d764-b2f3-498a-b16e-1a4d2a369dfa</SubscriberObjectId>
  <UserURI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa</UserURI>
  <ObjectId>0f4ce029-1939-4ae0-9dee-9d84c77fadae</ObjectId>
  <DisplayName>Home Phone</DisplayName>
  <Active>>false</Active>
  <BusyRetryInterval>5</BusyRetryInterval>
  <Conversation>SubNotify</Conversation>
  <Type>1</Type>

```

```

<DialDelay>1</DialDelay>
<MaxBody>512</MaxBody>
<MaxSubject>64</MaxSubject>
<RetriesOnBusy>4</RetriesOnBusy>
<RetriesOnRna>4</RetriesOnRna>
<RingsToWait>4</RingsToWait>
<RnaRetryInterval>15</RnaRetryInterval>
<SendCount>>false</SendCount>
<WaitConnect>>true</WaitConnect>
<MediaSwitchObjectId>6ecaa89b-2d29-4a21-8013-75c154ee58f5</MediaSwitchObjectId>

<PhoneSystemURI>/vmrest/phonesystems/6ecaa89b-2d29-4a21-8013-75c154ee58f5</PhoneSystemURI>

<TransmitForcedAuthorizationCode>>false</TransmitForcedAuthorizationCode>
<DeviceName>Home Phone</DeviceName>
<PromptForId>>false</PromptForId>
<SendCallerId>>false</SendCallerId>
<SendPcaLink>>false</SendPcaLink>
<Undeletable>>true</Undeletable>
<MediaSwitchDisplayName>PhoneSystem</MediaSwitchDisplayName>
<DetectTransferLoop>>false</DetectTransferLoop>
<SuccessRetryInterval>0</SuccessRetryInterval>
<RetriesOnSuccess>0</RetriesOnSuccess>
<EventList>NewVoiceMail</EventList>
<ScheduleSetObjectId>d7ea64ca-e3fc-4d0a-8bd5-ef2337a31b97</ScheduleSetObjectId>
<InitialDelay>0</InitialDelay>
<RepeatInterval>0</RepeatInterval>
<RepeatNotify>>false</RepeatNotify>
</NotificationDevice>
<NotificationDevice>

<URI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa/notificationdevices/878de0bf-9169-4ef7-afe4-ec82998c2322</URI>

<SubscriberObjectId>70c5d764-b2f3-498a-b16e-1a4d2a369dfa</SubscriberObjectId>
<UserURI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa</UserURI>
<ObjectId>878de0bf-9169-4ef7-afe4-ec82998c2322</ObjectId>
<DisplayName>Mobile Phone</DisplayName>
<Active>>false</Active>
<BusyRetryInterval>5</BusyRetryInterval>
<Conversation>SubNotify</Conversation>
<Type>1</Type>
<DialDelay>1</DialDelay>
<MaxBody>512</MaxBody>
<MaxSubject>64</MaxSubject>
<RetriesOnBusy>4</RetriesOnBusy>
<RetriesOnRna>4</RetriesOnRna>
<RingsToWait>4</RingsToWait>
<RnaRetryInterval>15</RnaRetryInterval>
<SendCount>>false</SendCount>
<WaitConnect>>true</WaitConnect>
<MediaSwitchObjectId>6ecaa89b-2d29-4a21-8013-75c154ee58f5</MediaSwitchObjectId>

<PhoneSystemURI>/vmrest/phonesystems/6ecaa89b-2d29-4a21-8013-75c154ee58f5</PhoneSystemURI>

<TransmitForcedAuthorizationCode>>false</TransmitForcedAuthorizationCode>
<DeviceName>Mobile Phone</DeviceName>
<PromptForId>>false</PromptForId>
<SendCallerId>>false</SendCallerId>
<SendPcaLink>>false</SendPcaLink>
<Undeletable>>true</Undeletable>
<MediaSwitchDisplayName>PhoneSystem</MediaSwitchDisplayName>
<DetectTransferLoop>>false</DetectTransferLoop>
<SuccessRetryInterval>0</SuccessRetryInterval>
<RetriesOnSuccess>0</RetriesOnSuccess>

```

```

    <EventList>NewVoiceMail</EventList>
    <ScheduleSetObjectId>d7ea64ca-e3fc-4d0a-8bd5-ef2337a31b97</ScheduleSetObjectId>
    <InitialDelay>0</InitialDelay>
    <RepeatInterval>0</RepeatInterval>
    <RepeatNotify>>false</RepeatNotify>
  </NotificationDevice>
</NotificationDevice>

<URI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa/notificationdevices/3a314c33-a69f-4179-a260-0c84a4520b3b</URI>

    <SubscriberObjectId>70c5d764-b2f3-498a-b16e-1a4d2a369dfa</SubscriberObjectId>
    <UserURI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa</UserURI>
    <ObjectId>3a314c33-a69f-4179-a260-0c84a4520b3b</ObjectId>
    <DisplayName>Work Phone</DisplayName>
    <Active>>true</Active>
    <AfterDialDigits>1701</AfterDialDigits>
    <BusyRetryInterval>5</BusyRetryInterval>
    <Conversation>SubNotify</Conversation>
    <Type>1</Type>
    <DialDelay>1</DialDelay>
    <MaxBody>512</MaxBody>
    <MaxSubject>64</MaxSubject>
    <PhoneNumber>3475</PhoneNumber>
    <RetriesOnBusy>4</RetriesOnBusy>
    <RetriesOnRna>4</RetriesOnRna>
    <RingsToWait>4</RingsToWait>
    <RnaRetryInterval>15</RnaRetryInterval>
    <SendCount>>false</SendCount>
    <WaitConnect>>false</WaitConnect>
    <MediaSwitchObjectId>6ecaa89b-2d29-4a21-8013-75c154ee58f5</MediaSwitchObjectId>

  <PhoneSystemURI>/vmrest/phonesystems/6ecaa89b-2d29-4a21-8013-75c154ee58f5</PhoneSystemURI>

    <TransmitForcedAuthorizationCode>>false</TransmitForcedAuthorizationCode>
    <DeviceName>Work Phone</DeviceName>
    <PromptForId>>true</PromptForId>
    <SendCallerId>>false</SendCallerId>
    <SendPcaLink>>false</SendPcaLink>
    <Undeletable>>true</Undeletable>
    <MediaSwitchDisplayName>PhoneSystem</MediaSwitchDisplayName>
    <DetectTransferLoop>>false</DetectTransferLoop>
    <SuccessRetryInterval>0</SuccessRetryInterval>
    <RetriesOnSuccess>0</RetriesOnSuccess>
    <EventList>AllMessage</EventList>
    <ScheduleSetObjectId>d7ea64ca-e3fc-4d0a-8bd5-ef2337a31b97</ScheduleSetObjectId>
    <InitialDelay>5</InitialDelay>
    <RepeatInterval>15</RepeatInterval>
    <RepeatNotify>>true</RepeatNotify>
    <FailDeviceObjectId>0f4ce029-1939-4ae0-9dee-9d84c77fadae</FailDeviceObjectId>
  </NotificationDevice>
</NotificationDevice>

<URI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa/notificationdevices/3e7f802b-d707-489f-b241-399ab67a0dc3</URI>

    <SubscriberObjectId>70c5d764-b2f3-498a-b16e-1a4d2a369dfa</SubscriberObjectId>
    <UserURI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa</UserURI>
    <ObjectId>3e7f802b-d707-489f-b241-399ab67a0dc3</ObjectId>
    <DisplayName>SMTP</DisplayName>
    <Active>>false</Active>
    <BusyRetryInterval>0</BusyRetryInterval>
    <Type>4</Type>
    <DialDelay>0</DialDelay>
    <MaxBody>512</MaxBody>
    <MaxSubject>64</MaxSubject>

```

```

<RetriesOnBusy>0</RetriesOnBusy>
<RetriesOnRna>0</RetriesOnRna>
<RingsToWait>0</RingsToWait>
<RnaRetryInterval>0</RnaRetryInterval>
<SendCount>true</SendCount>
<WaitConnect>false</WaitConnect>
<TransmitForcedAuthorizationCode>false</TransmitForcedAuthorizationCode>
<DeviceName>SMTP</DeviceName>
<PromptForId>false</PromptForId>
<SendCallerId>true</SendCallerId>
<SendPcaLink>false</SendPcaLink>
<Undeletable>true</Undeletable>
<DetectTransferLoop>false</DetectTransferLoop>
<SuccessRetryInterval>0</SuccessRetryInterval>
<RetriesOnSuccess>0</RetriesOnSuccess>
<EventList>NewVoiceMail</EventList>
<ScheduleSetObjectId>d7ea64ca-e3fc-4d0a-8bd5-ef2337a31b97</ScheduleSetObjectId>
<InitialDelay>0</InitialDelay>
<RepeatInterval>0</RepeatInterval>
<RepeatNotify>false</RepeatNotify>
</NotificationDevice>
</NotificationDevices>

```

Listing and Viewing Specialized Devices

The following is an example of a GET that lists all Phone Devices for the specified User. Listing all specialized devices of other types is very similar; simply do a GET to the pagerdevices, smsdevices, or smtpdevices resource, rather than to the phonedevices resource.

```
GET https://<connection-server>/vmrest/users/<UserObjectId>/notificationdevices/phonedevices
```

The following is the response from the above GET request. Note that each returned resource is a Phone Device rather than a Notification Device since the request was for Phone Devices only.


```

200
OK

<?xml version="1.0" encoding="UTF-8"?>
<PhoneDevices total="3">
  <PhoneDevice>

<URL>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa/notificationdevices/phonedevices/3a314c33-a69f-4179-a260-0c84a4520b3b</URL>

  <TransmitForcedAuthorizationCode>>false</TransmitForcedAuthorizationCode>
  <AfterDialDigits>1701</AfterDialDigits>
  <BusyRetryInterval>5</BusyRetryInterval>
  <Conversation>SubNotify</Conversation>
  <DialDelay>1</DialDelay>
  <PhoneNumber>3475</PhoneNumber>
  <RetriesOnBusy>4</RetriesOnBusy>
  <RetriesOnRna>4</RetriesOnRna>
  <RingsToWait>4</RingsToWait>
  <RnaRetryInterval>15</RnaRetryInterval>
  <WaitConnect>>false</WaitConnect>
  <MediaSwitchObjectId>6ecaa89b-2d29-4a21-8013-75c154ee58f5</MediaSwitchObjectId>

<PhoneSystemURI>/vmrest/phonesystems/6ecaa89b-2d29-4a21-8013-75c154ee58f5</PhoneSystemURI>

  <ObjectId>3a314c33-a69f-4179-a260-0c84a4520b3b</ObjectId>
  <Active>>true</Active>
  <DeviceName>Work Phone</DeviceName>
  <DisplayName>Work Phone</DisplayName>
  <MaxBody>512</MaxBody>
  <MaxSubject>64</MaxSubject>
  <SubscriberObjectId>70c5d764-b2f3-498a-b16e-1a4d2a369dfa</SubscriberObjectId>
  <UserURI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa</UserURI>
  <PromptForId>>true</PromptForId>
  <Undeletable>>true</Undeletable>
  <DetectTransferLoop>>false</DetectTransferLoop>
  <EventList>AllMessage</EventList>
  <ScheduleSetObjectId>d7ea64ca-e3fc-4d0a-8bd5-ef2337a31b97</ScheduleSetObjectId>
  <InitialDelay>5</InitialDelay>
  <RepeatInterval>15</RepeatInterval>
  <RepeatNotify>true</RepeatNotify>
  <FailDeviceObjectId>0f4ce029-1939-4ae0-9dee-9d84c77fadae</FailDeviceObjectId>
</PhoneDevice>
<PhoneDevice>

<URL>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa/notificationdevices/phonedevices/0f4ce029-1939-4ae0-9dee-9d84c77fadae</URL>

  <TransmitForcedAuthorizationCode>>false</TransmitForcedAuthorizationCode>
  <BusyRetryInterval>5</BusyRetryInterval>
  <Conversation>SubNotify</Conversation>
  <DialDelay>1</DialDelay>
  <RetriesOnBusy>4</RetriesOnBusy>
  <RetriesOnRna>4</RetriesOnRna>
  <RingsToWait>4</RingsToWait>
  <RnaRetryInterval>15</RnaRetryInterval>
  <WaitConnect>true</WaitConnect>
  <MediaSwitchObjectId>6ecaa89b-2d29-4a21-8013-75c154ee58f5</MediaSwitchObjectId>

<PhoneSystemURI>/vmrest/phonesystems/6ecaa89b-2d29-4a21-8013-75c154ee58f5</PhoneSystemURI>

  <ObjectId>0f4ce029-1939-4ae0-9dee-9d84c77fadae</ObjectId>
  <Active>>false</Active>
  <DeviceName>Home Phone</DeviceName>
  <DisplayName>Home Phone</DisplayName>
  <MaxBody>512</MaxBody>

```

Listing of HTML Notification Device for a User (in Cisco Unity Connection 9.0(1) and Later)

```

<MaxSubject>64</MaxSubject>
<SubscriberObjectId>70c5d764-b2f3-498a-b16e-1a4d2a369dfa</SubscriberObjectId>
<UserURI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa</UserURI>
<PromptForId>>false</PromptForId>
<Undeletable>>true</Undeletable>
<DetectTransferLoop>>false</DetectTransferLoop>
<EventList>NewVoiceMail</EventList>
<ScheduleSetObjectId>d7ea64ca-e3fc-4d0a-8bd5-ef2337a31b97</ScheduleSetObjectId>
<InitialDelay>0</InitialDelay>
<RepeatInterval>0</RepeatInterval>
<RepeatNotify>>false</RepeatNotify>
</PhoneDevice>
<PhoneDevice>
<URL>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa/notificationdevices/phonedevice/878de0bf-9169-4ef7-afe4-ec82998c2322</URL>
<TransmitForcedAuthorizationCode>>false</TransmitForcedAuthorizationCode>
<BusyRetryInterval>5</BusyRetryInterval>
<Conversation>SubNotify</Conversation>
<DialDelay>1</DialDelay>
<RetriesOnBusy>4</RetriesOnBusy>
<RetriesOnRna>4</RetriesOnRna>
<RingsToWait>4</RingsToWait>
<RnaRetryInterval>15</RnaRetryInterval>
<WaitConnect>>true</WaitConnect>
<MediaSwitchObjectId>6ecaa89b-2d29-4a21-8013-75c154ee58f5</MediaSwitchObjectId>
<PhoneSystemURI>/vmrest/phonesystems/6ecaa89b-2d29-4a21-8013-75c154ee58f5</PhoneSystemURI>
<ObjectId>878de0bf-9169-4ef7-afe4-ec82998c2322</ObjectId>
<Active>>false</Active>
<DeviceName>Mobile Phone</DeviceName>
<DisplayName>Mobile Phone</DisplayName>
<MaxBody>512</MaxBody>
<MaxSubject>64</MaxSubject>
<SubscriberObjectId>70c5d764-b2f3-498a-b16e-1a4d2a369dfa</SubscriberObjectId>
<UserURI>/vmrest/users/70c5d764-b2f3-498a-b16e-1a4d2a369dfa</UserURI>
<PromptForId>>false</PromptForId>
<Undeletable>>true</Undeletable>
<DetectTransferLoop>>false</DetectTransferLoop>
<EventList>NewVoiceMail</EventList>
<ScheduleSetObjectId>d7ea64ca-e3fc-4d0a-8bd5-ef2337a31b97</ScheduleSetObjectId>
<InitialDelay>0</InitialDelay>
<RepeatInterval>0</RepeatInterval>
<RepeatNotify>>false</RepeatNotify>
</PhoneDevice>
/PhoneDevices>

```

Listing of HTML Notification Device for a User (in Cisco Unity Connection 9.0(1) and Later)

The following is an example of the GET request that lists the HTML notification devices for a user:

```
GET /vmrest/users/<userobjectid>/notificationdevices/htmldevices
```

The following is the response from the above *GET* request:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <HtmlDevices total="2">
    <HtmlDevice>

<URL>/vmrest/users/24cb82fc-5153-4c3f-abc7-ef442b27b4e9/notificationdevices/htmldevices/eb8af2c7-1466-4d36-9f8a-957c001ad4c5</URL>

    <Active>>false</Active>
    <CallbackNumber></CallbackNumber>
    <DeviceName>HTML</DeviceName>
    <DisableMobileNumberFromPCA>>false</DisableMobileNumberFromPCA>
    <DisplayName>HTML</DisplayName>
    <NotificationTemplateID>a00ea2a4-623e-414e-bf9c-a15c77075766</NotificationTemplateID>
    <ObjectId>eb8af2c7-1466-4d36-9f8a-957c001ad4c5</ObjectId>
    <PhoneNumber></PhoneNumber>
    <SmtpAddress></SmtpAddress>
    <Undeletable>>true</Undeletable>
    <SubscriberObjectId>24cb82fc-5153-4c3f-abc7-ef442b27b4e9</SubscriberObjectId>
    <UserURI>/vmrest/users/24cb82fc-5153-4c3f-abc7-ef442b27b4e9</UserURI>
    <EventList>NewVoiceMail</EventList>
    <ScheduleSetObjectId>e6706d47-257f-4e24-bf9f-a182bb3b2633</ScheduleSetObjectId>
    <InitialDelay>0</InitialDelay>
    <RepeatInterval>0</RepeatInterval>
    <RepeatNotify>>false</RepeatNotify>
  </HtmlDevice>
  <HtmlDevice>

<URL>/vmrest/users/24cb82fc-5153-4c3f-abc7-ef442b27b4e9/notificationdevices/htmldevices/64d260ca-7f09-46df-be87-88cd3ba9c5e3</URL>

    <Active>>false</Active>
    <DeviceName>SMTP</DeviceName>
    <DisableMobileNumberFromPCA>>true</DisableMobileNumberFromPCA>
    <DisplayName>HTML_3</DisplayName>
    <NotificationTemplateID>103c4514-db6e-489b-bc08-e9e131b60099</NotificationTemplateID>
    <ObjectId>64d260ca-7f09-46df-be87-88cd3ba9c5e3</ObjectId>
    <Undeletable>>true</Undeletable>
    <SubscriberObjectId>24cb82fc-5153-4c3f-abc7-ef442b27b4e9</SubscriberObjectId>
    <UserURI>/vmrest/users/24cb82fc-5153-4c3f-abc7-ef442b27b4e9</UserURI>
    <EventList>NewVoiceMail</EventList>
    <ScheduleSetObjectId>e6706d47-257f-4e24-bf9f-a182bb3b2633</ScheduleSetObjectId>
    <InitialDelay>0</InitialDelay>
    <RepeatInterval>0</RepeatInterval>
    <RepeatNotify>>false</RepeatNotify>
  </HtmlDevice>
</HtmlDevices>

```

Listing Properties of an HTML Notification Device (in Cisco Unity Connection 9.0(1) and Later)

The following is an example of the GET request that displays the properties for a particular HTML notification device:

```
GET /vmrest/users/<userobjectid>/notificationdevices/htmldevices/<deviceid>
```

The following is the response from the above *GET* request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <HtmlDevice>

<URL>/vmrest/users/24cb82fc-5153-4c3f-abc7-ef442b27b4e9/notificationdevices/htmldevices/64d260ca-7f09-46df-be87-88cd3ba9c5e3</URL>

  <Active>>false</Active>
  <DeviceName>SMTP</DeviceName>
  <DisableMobileNumberFromPCA>>true</DisableMobileNumberFromPCA>
  <DisplayName>HTML_3</DisplayName>
  <NotificationTemplateID>103c4514-db6e-489b-bc08-e9e131b60099</NotificationTemplateID>
  <ObjectId>64d260ca-7f09-46df-be87-88cd3ba9c5e3</ObjectId>
  <Undeletable>>true</Undeletable>
  <SubscriberObjectId>24cb82fc-5153-4c3f-abc7-ef442b27b4e9</SubscriberObjectId>
  <UserURI>/vmrest/users/24cb82fc-5153-4c3f-abc7-ef442b27b4e9</UserURI>
  <EventList>NewVoiceMail</EventList>
  <ScheduleSetObjectId>e6706d47-257f-4e24-bf9f-a182bb3b2633</ScheduleSetObjectId>
  <InitialDelay>0</InitialDelay>
  <RepeatInterval>0</RepeatInterval>
  <RepeatNotify>>false</RepeatNotify>
</HtmlDevice>
```

Searching

To retrieve a list of Notification Devices that meet a specified search criteria, add the following query parameter to a GET: query=(column [is | startswith] value)

For example, to find all Notification Devices with a DisplayName that starts with "Phone":

```
GET
https://<connection-server>/vmrest/users/<UserObjectId/notificationdevices?query=(DisplayName%20startswith%20Home)
```

Search criteria can also be used when doing a GET to the specialized Device resources.

Creating

A generic Notification Device resource is read-only, so new ones cannot be created. However, the four specialized Device resources are not read-only, so new ones can be created. Each of the four specialized Device resources has different required parameters during creation.

Updating

A generic Notification Device resource is read-only, so Notification Devices cannot be updated. However, the five specialized Device resources are not read-only, so an API user can update the appropriate specialized Device resource.

Updating an Existing Phone Device

The following is an example of a PUT request that modifies the FailDeviceObjectId of an existing Phone Device. The FailDeviceObjectId indicates which Notification Device to try if notification fails to the current device, which allows you to setup a chain of Notification Devices.

```

PUT
https://<connection-server>/vmrest/users/<UserObjectId>/notificationdevices/phonedevicess/2e377eba-a924-434b-b967-bac3815551d3

<PhoneDevice>
  <FailDeviceObjectId>ac200457-284c-4951-b04a-4d76bd03235e</FailDeviceObjectId>
</PhoneDevice>

```

The following is the response from the above PUT request:

```

204
No Content

```

Updating other types of specialized Device resources is very similar; simply do a PUT to the appropriate pagerdevice, smsdevice, or smtpdevice resource instead of a phonedevice resource.

Updating an HTML Notification Device to Apply the HTML Notification Template (in Cisco Unity Connection 9.0(1) and Later)

The following is an example of the PUT request that modifies the HTML notification device as represented by <deviceid>:

```

PUT /vmrest/users/<userobjectid>/notificationdevices/htmldevices/<deviceid>

<HtmlDevice>
  <NotificationTemplateID>103c4514-db6e-489b-bc08-e9e131b60099</NotificationTemplateID>
</HtmlDevice>

```

The input for the PUT request will be XML or JSON as per HtmlDevice schema. The output for this request returns the successful response code.

Deleting

A generic Notification Device resource is read-only, so Notification Devices cannot be deleted. However, the four specialized Device resources are not read-only, so an API user can delete the appropriate specialized Device resource, assuming its not one of the factory default Device resources (which are flagged as Undeletable).

Explanation of Data Fields

The following chart lists all of the data fields available on Notification Devices, Phone Devices, Pager Devices, SMS Devices, and SMTP Devices. In the Device Types column, "Base" means the generic Notification Device resource. Note that on a GET to the generic Notification Device resource, you may receive data fields that are meaningless for a particular type of device, so these fields can be safely ignored.

Field Name	Device Type	Writable?	Explanation/Comments
ObjectId	All	Read Only	ObjectId of the Device
Type	Base	Read Only	Auto-set during Create. 1=Phone, 2=Pager, 4=SMTP, 5=SMS
Active	All	Read/Write	Factory default=false. For newly-created, default=true

Field Name	Device Type	Writable?	Explanation/Comments
AfterDialDigits	Base, Phone, Pager	Read/Write	Extra-digits to dial after the Phone Number. Default=null
BusyRetryInterval	Base, Phone, Pager	Read/Write	Time in minutes to wait between retries if Busy. Default=5
Conversation	Base, Phone	Read Only	Auto-set during Create.
DetectTransferLoop	Base, Phone	Read/Write	Flags if Transfer Loop Detection is Enabled. Default=false
DeviceName	All	Read/Write	Friendly name for the Device type (not shown in CUCA or CPCA). Default="Other"
DialDelay	Base, Phone, Pager	Read/Write	Time in seconds to wait after Connect before dialing AfterDialDigits (ignored if no AfterDialDigits). Default=1
DisplayName	All	Read/Write	Friendly name for the Device like "Mobile Phone"
EventList	All	Read/Write	Comma-delimited list of Events that trigger Notification (see below). Default=NewVoiceMail
FailDeviceObjectid	All	Read/Write	ObjectId of next Notification Device to use if this one fails (to chain Notifications). Default=null
InitialDelay	All	Read/Write	Time in minutes to wait after a Message is received before Notification is triggered. Default=0
MaxBody	All	Read/Write	Maximum number of characters in the Body of a Notification message. Default=512
MaxSubject	All	Read/Write	Maximum number of characters in the Subject of a Notification message. Default=64
MediaSwitchObjectId	Base, Phone, Pager	Read/Write	ObjectId of the Phone System to use for the Dial-out.
PhoneNumber	Base, Phone, Pager	Read/Write	Phone Number to dial to reach the Device.
PromptForId	Base, Phone	Read/Write	Flags if the User must enter their ID to get the Notification message. Default=false
RecipientAddress	SMS	Read/Write	Address of Message recipient
RepeatInterval	All	Read/Write	Time in minutes to wait before re-notifying of messages. Default=0
RepeatNotify	All	Read/Write	Flags if Notification process should begin for each newly arrived message. Default=false
RetriesOnBusy	Base, Phone, Pager	Read/Write	Number of retry attempts if Busy. Default=4
RetriesOnRna	Base, Phone, Pager	Read/Write	Number of retry attempts if RNA. Default=4
RetriesOnSuccess	Base, Phone	Read/Write	Number of retry attempts if successful. Default=0
RingsToWait	Base, Phone, Pager	Read/Write	Number of rings to wait for Connect before RNA. Default=4

Field Name	Device Type	Writable?	Explanation/Comments
RnaRetryInterval	Base, Phone, Pager	Read/Write	Time in minutes to wait between retries if RNA. Default=5
ScheduleSetObjectId	All	Read/Write	ObjectId of the ScheduleSet during which Notification may trigger. Default=AllHours
SendCallerId	Base, Pager, SMS, SMTP	Read/Write	Flag to include CallerId in the Notification message. Default=true
SendCount	Base, Pager, SMS, SMTP	Read/Write	Flag to include the message counts in the Notification message. Default=true
SenderAddress	SMS	Read/Write	Address of Message sender. Default=null
SendPcaLink	Base, SMTP	Read/Write	Flag to include a link to CPCA in the Notification message. Default=false
SmppProviderObjectId	Base, SMS	Read/Write	ObjectId of the SMPP Provider
SmtppAddress	Base, SMTP	Read/Write	SMTP address to send the Notification message to
StaticText	Base, SMS, SMTP	Read/Write	Static text to include in the Notification Message. Default=null
SubscriberObjectId	All	Read Only	ObjectId of the User
SuccessRetryInterval	Base, Pager	Read/Write	Time in minutes to wait between retries if successful. Default=0
TransmitAuthorizationCode	Base, Phone, Pager	Read/Write	Flags if an authorization code should be sent to CUCM after the PhoneNumber is dialed. Default=false
Undeletable	All	Read Only	Factory default=true. For newly-created, default=false
WaitConnect	Base, Phone, Pager	Read/Write	Flags if need to wait for Connect before dialing AfterDialDigits. Default=true

The EventList field is a comma-delimited list of Message types which could trigger Notification. It can contain any of the following values. Note that Calendar Appointments and Meetings are not included in the AllMessage or AllUrgentMessage categories, which just include voicemail, fax, and dispatch messages.

Event Name	Description
None	No messages of any type or urgency
AllMessage	All messages of all types and urgencies
AllUrgentMessage	All urgent messages of all types
CalendarAppointment	Calendar appointments
CalendarMeeting	Calendar meetings
DispatchMessage	All dispatch messages
UrgentDispatchMessage	All urgent dispatch messages

Event Name	Description
NewFax	All fax messages
NewUrgentFax	All urgent fax messages
NewVoiceMail	All voicemail messages
NewUrgentVoiceMail	All urgent voicemail messages

Cisco Unity Connection Provisioning Interface (CUPI) API -- Private List Members

Viewing

Every private list has members. To see the members of a private list, do the following GET:

```
GET /vmrest/users/<userobjectid>/privatelists/<privatelistobjectid>/privatelistmembers
```

Here is an example of the response you might receive to such a request:


```

200
OK
<?xml version="1.0" encoding="UTF-8"?>
<PrivateListMembers total="4">
  <PrivateListMember>
    <URI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928/privatelists/
df523f3a-4e40-4d6d-9f80-72a5bc93331c/privatelistmembers/
4d390786-56a8-45e3-85a1-1c0933375b99</URI>

<PersonalVoiceMailListObjectId>df523f3a-4e40-4d6d-9f80-72a5bc93331c</PersonalVoiceMailListObjectId>

    <PrivateListURI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928/
privatelists/df523f3a-4e40-4d6d-9f80-72a5bc93331c</PrivateListURI>
    <MemberSubscriberObjectId>444ded7b-e2b9-48c1-9aca-bcfda4372928</
MemberSubscriberObjectId>
    <MemberUserURI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928</
MemberUserURI>
    <ObjectId>4d390786-56a8-45e3-85a1-1c0933375b99</ObjectId>
    <Alias>UserA</Alias>
    <DisplayName>User A</DisplayName>
    <Extension>1017</Extension>
  </PrivateListMember>
  <PrivateListMember>
    <URI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928/privatelists/
df523f3a-4e40-4d6d-9f80-72a5bc93331c/privatelistmembers/
4e16bf12-6383-46d9-ae32-9ca9a857ca4f</URI>

<PersonalVoiceMailListObjectId>df523f3a-4e40-4d6d-9f80-72a5bc93331c</PersonalVoiceMailListObjectId>

    <PrivateListURI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928/
privatelists/df523f3a-4e40-4d6d-9f80-72a5bc93331c</PrivateListURI>

<MemberDistributionListObjectId>4523ba2f-015c-414e-ad96-92aba431ed39</MemberDistributionListObjectId>

    <MemberDistributionListURI>/vmrest/distributionlists/
4523ba2f-015c-414e-ad96-92aba431ed39</MemberDistributionListURI>
    <ObjectId>4e16bf12-6383-46d9-ae32-9ca9a857ca4f</ObjectId>
    <Alias>ListA</Alias>
    <DisplayName>List A</DisplayName>
  </PrivateListMember>
  <PrivateListMember>
    <URI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928/privatelists/
df523f3a-4e40-4d6d-9f80-72a5bc93331c/privatelistmembers/
c61a60f6-0aa5-4c85-94ba-94fd6c167bd4</URI>

<PersonalVoiceMailListObjectId>df523f3a-4e40-4d6d-9f80-72a5bc93331c</PersonalVoiceMailListObjectId>

    <PrivateListURI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928/
privatelists/df523f3a-4e40-4d6d-9f80-72a5bc93331c</PrivateListURI>

<MemberPersonalVoiceMailListObjectId>48eb2b0f-15a8-4bdf-a3a0-098debed181d</MemberPersonalVoiceMailListObjectId>

    <MemberPrivateListURI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928/
privatelists/48eb2b0f-15a8-4bdf-a3a0-098debed181d</MemberPrivateListURI>
    <ObjectId>c61a60f6-0aa5-4c85-94ba-94fd6c167bd4</ObjectId>
    <Alias>UserA_48eb2b0f-15a8-4bdf-a3a0-098debed181d</Alias>
    <DisplayName>2</DisplayName>
  </PrivateListMember>
  <PrivateListMember>
    <URI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928/privatelists/
df523f3a-4e40-4d6d-9f80-72a5bc93331c/privatelistmembers/
41295a7c-c9c0-45d3-8060-5a5038f0a49e</URI>

<PersonalVoiceMailListObjectId>df523f3a-4e40-4d6d-9f80-72a5bc93331c</PersonalVoiceMailListObjectId>

```

```

    <PrivateListURI>/vmrest/users/444ded7b-e2b9-48c1-9aca-bcfda4372928/
privatelists/df523f3a-4e40-4d6d-9f80-72a5bc93331c</PrivateListURI>
    <ObjectId>41295a7c-c9c0-45d3-8060-5a5038f0a49e</ObjectId>
    <MemberContactObjectId>27e58431-981f-4fbd-b667-69b1aaac89fd</
MemberContactObjectId>
    <MemberContactURI>/vmrest/contacts/27e58431-981f-4fbd-b667-69b1aaac89fd</
MemberContactURI>
    <Alias>ConA</Alias>
    <DisplayName>Con A</DisplayName>
    <Extension>1290390</Extension>
  </PrivateListMember>
</PrivateListMembers>

```

Every member of a private list can be one of four objects: a user, a distribution list, another private list, or a remote contact (a contact that can receive a message). Depending on which object the member is, different fields are used to track the member information, which consists solely of an object id and a URI to that object.

The Alias, Display Name, and Extension of the member in question are also included. These fields are not actual values on the private list member data object; instead, the values are looked up by the API when it performs the GET.

Creating

The following POST adds a member to a private list. In this example, a user is being added to the private list by simply providing the object id of the user.

```

POST /vmrest/users/<userobjectid>/privatelists/<privatelistobjectid>/
privatelistmembers
<?xml version="1.0" encoding="UTF-8"?>
<PrivateListMember>
  <MemberSubscriberObjectId>9bc04f85-9ac4-42f8-9314-547408a6126c</
MemberSubscriberObjectId>
</PrivateListMember>

```

Use `MemberDistributionListObjectId` to add distribution lists to a private list. Use `MemberPersonalVoiceMailListObjectId` to add other private lists, and use `MemberContactObjectId` to add contacts.

Deleting

The following DELETE removes a member from a private list:

```

DELETE /vmrest/users/<userobjectid>/privatelists/<privatelistobjectid>/
privatelistmembers/<privatelistmemberobjectid>

```

Cisco Unity Connection Provisioning Interface (CUPI) API -- Private Lists

Viewing

Every user with a mailbox can have private lists. To see the private lists belonging to a user, do the following GET request:

```
GET /vmrest/users/<user object id>/privatelists/
```

Here is an example of the response you might receive to such a request:

```
<?xml version="1.0" encoding="UTF-8"?>
<PrivateLists total="1">
  <PrivateList>
    <URI>/vmrest/users/f90d26fc-0e4a-405a-8c42-189a93129bdc/
privatelists/1f443e29-1d6b-4ef8-89a6-2767549e3577</URI>
    <ObjectId>1f443e29-1d6b-4ef8-89a6-2767549e3577</ObjectId>
    <DisplayName>1</DisplayName>
    <UserObjectId>f90d26fc-0e4a-405a-8c42-189a93129bdc</UserObjectId>
    <UserURI>/vmrest/users/f90d26fc-0e4a-405a-8c42-189a93129bdc</UserURI>
    <DtmfName>1</DtmfName>
    <Alias>UserA_1f443e29-1d6b-4ef8-89a6-2767549e3577</Alias>
    <VoiceName>6ce746b7-d775-4bc4-9d19-8b6a55b9461d.wav</VoiceName>
    <VoiceFileURI>/vmrest/voicefiles/
6ce746b7-d775-4bc4-9d19-8b6a55b9461d.wav</VoiceFileURI>
    <NumericId>1</NumericId>
    <IsAddressable>true</IsAddressable>
  </PrivateList>
</PrivateLists>
```

Many of these fields are consistent with what you see on other objects. Specific fields that are unique to private lists and require explanation are described below.

Private lists belong to specific users. The **UserObjectId** and **UserURI** field show that relationship.

DtmfName is a field with a value that is automatically generated every time the display name changes on the private list. DtmfName holds the digits that would need to be dialed to address a message by name to this group by phone.

Alias is a unique text name for the private list. The alias is automatically generated, and to ensure it is unique, it is constructed by concatenating the user alias, an underscore and the private list's object id. The user's alias is truncated to 28 characters to avoid overflow.

NumericId is the number of the private list. Private lists are numbered 1 through 99, and when addressing a message by extension over the phone, the number can be used to choose the group.

IsAddressable is an odd flag. It flags whether or not the private list is addressable. By default, all private lists are addressable.

Creating

The following POST creates a private list. In this example, it would be private list 2.

```
POST /vmrest/users/<user object id>/privatelists

<?xml version="1.0" encoding="UTF-8"?>
<PrivateList>
  <DisplayName>2</DisplayName>
  <VoiceName>6ce746b7-d775-4bc4-9d19-8b6a55b9461d.wav</VoiceName>
  <NumericId>2</NumericId>
</PrivateList>
```

These are the only three parameters that can be explicitly set on a new private list. The other parameters are not allowed on a POST, either defaulting to a certain value or being auto-generated as discussed above. The ObjectId can be explicitly set if desired, but that is optional.

Updating

Only the following fields can be updated in a PUT request:

- DisplayName
- VoiceName
- NumericId

The following example shows a PUT request that modifies some fields, changing a private list to private list 6.

```
PUT /vmrest/users/<user object id>/privatelists/<private list object id>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<PrivateList>
  <DisplayName>6</DisplayName>
  <NumericId>6</NumericId>
</PrivateList>
```

Deleting

A private list can be deleted by a DELETE request:

```
DELETE /vmrest/users/<user object id>/privatelists/<private list object id>
```

Voice Names

Creating a voice name for a private list (or for any object exposed in CUPI) is a multi-step process.

Step 1: Create a place-holder for the WAV file by doing a POST:

```
POST /vmrest/voicefiles
```

The name of the WAV file will be included in the response.

```
201
Created
6f9c6d05-42e5-48e3-8d07-204250af320d.wav
```

Step 2: Take the name returned in Step 1, and do a PUT where the HTTP content type is "audio/wav" and the payload content is the audio data:

```
PUT /vmrest/voicefiles/6f9c6d05-42e5-48e3-8d07-204250af320d.wav
```

Step 3: Take the name of the WAV file and do a PUT request on the private list whose voice name you want to set:

```
PUT /vmrest/users/<user object id>/privatelists/<private list object id>

<?xml version="1.0" encoding="UTF-8"?>
<PrivateList>
  <VoiceName>6ce746b7-d775-4bc4-9d19-8b6a55b9461d.wav</VoiceName>
</PrivateList>
```

The voice name can always be retrieved through the voice name URI, it will return the audio of the greeting as an "audio/wav" media type:

```
GET /vmrest/voicefiles/6f9c6d05-42e5-48e3-8d07-204250af320d.wav
```

Cisco Unity Connection Provisioning Interface (CUPI) API -- SMTP Proxy Addresses

About SMTP Proxy Addresses

This page contains information on how to use the API to list SMTP Proxy Addresses.

Listing and Viewing

The following is an example of a GET that lists all <element name>:

```
GET http://<connection-server>/vmrest/smtpproxyaddresses
```

The following is the response from the above GET request:

```

200
OK
<?xml version="1.0" encoding="UTF-8"?>
<SmtproxyAddresses total="2">
  <SmtproxyAddress>
    <URI>/vmrest/smtproxyaddresses/9fd21b87-1509-42f1-88ce-3f36122c68ee</URI>
    <ObjectId>9fd21b87-1509-42f1-88ce-3f36122c68ee</ObjectId>
    <SmtAddress>somedude@somewhere.com</SmtAddress>
    <ObjectGlobalUserObjectId>0804bda6-953c-4271-b44a-0830c1429af2</ObjectGlobalUserObjectId>

  <ObjectGlobalUserURI>/vmrest/globalusers/0804bda6-953c-4271-b44a-0830c1429af2</ObjectGlobalUserURI>

  </SmtproxyAddress>
  <SmtproxyAddress>
    <URI>/vmrest/smtproxyaddresses/fc107ad8-b9e5-409e-b0bc-62e295c7532e</URI>
    <ObjectId>fc107ad8-b9e5-409e-b0bc-62e295c7532e</ObjectId>
    <SmtAddress>someotherdude@somewhereelse.com</SmtAddress>
    <ObjectGlobalUserObjectId>0804bda6-953c-4271-b44a-0830c1429af2</ObjectGlobalUserObjectId>

  <ObjectGlobalUserURI>/vmrest/globalusers/0804bda6-953c-4271-b44a-0830c1429af2</ObjectGlobalUserURI>

  </SmtproxyAddress>
</SmtproxyAddresses>

```

To retrieve a sorted list of all SMTP Proxy Addresses, add the following query parameter: `sort=(column [asc | desc])`

For example, to retrieve a list of all SMTP Proxy Addresses sorted by SMTP Address in ascending order:

```
GET http://<connection-server>/vmrest/smtproxyaddresses?sort=(smtpaddress%20asc)
```

To retrieve a specific SMTP Proxy Address by its object ID:

```
GET http://<connection-server>/vmrest/smtproxyaddresses/<objectid>
```

An SMTP Proxy Address can belong to one of four different objects: a global user, a contact, a distribution list, or a private list. In the above example, both SMTP Proxy Addresses belong to a user because `ObjectGlobalUserObjectId` property is set. The SMTP Proxy Address belongs to a contact if the `ObjectContactObjectId` property is set, a distribution list if the `ObjectDistributionListObjectId` property is set, or a private list if the `ObjectPersonalGroupObjectId` is set.

Searching

To retrieve a list of SMTP Proxy Addresses that meet a specified search criteria, add the following query parameter to a GET: `query=(column [is | startswith] value)`

For example, to find all SMTP Proxy Addresses with an SMTP Address that starts with "a":

```
GET http://<connection-server>/vmrest/smtproxyaddresses?query=(smtpaddress%20startswith%20a)
```

The following properties can be used for searching and sorting SMTP Proxy Addresses:

- SmtAddress
- ObjectId
- ObjectContactObjectId

- ObjectDistributionListObjectId
- ObjectGlobalUserObjectId
- ObjectPersonalGroupObjectId

Cisco Unity Connection Provisioning Interface (CUPI) API -- User Import

LDAP Import

A GET of an LDAP user will have a phone number. When a user is imported, an LDAP import must specify a dtmfAccessId in the payload (this is often based on the phone number).

In this document, <type> in a URI refers to ldap value.

Initial Configuration

Before any LDAP users can be imported, the LDAP synchronization has to be set as described in the [LDAP Directory Integration with Cisco Unity Connection 10.x](https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/10x/design/guide/10xcucdgx.html) chapter of *Design Guide for Cisco Unity Connection Release 10.x* available at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/10x/design/guide/10xcucdgx.html.

GET a List of Users That Can Be Imported

```
GET http://<server>/vmrest/import/users/<type>
```

Only users that are eligible for import are returned; users that are already imported are not in this list.

Offset and Limit Parameters Can Be Part of the Ldap Query

```
http://<server>/vmrest/import/users/<type>?offset=x&limit=y
```

Note the following:

- Either parameter can be specified individually
- If the offset exceeds the number of entries, the response is an empty list
- If no limit is specified and more than 2000 results are returned, the response is an error

Filter and Sort

To put constraints on search results, LDAP requests support the CUPI standard filter and sort parameters.

Example

```
GET /vmrest/import/users/ldap?limit=5 HTTP/1.1
Accept: application/json
User-Agent: Java/1.6.0_20
Host: cuc-install-69.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
```

The GET above would produce the following response:

```
HTTP/1.1 200 OK
Pragma: No-cache
Cache-Control: no-cache
Expires: Wed, 31 Dec 1969 16:00:00 PST
Set-Cookie: JSESSIONIDSSO=11D3599617B30496BAD4BF2BB4C23B32; Path=/
Set-Cookie: JSESSIONID=2D9E4ACB334EF6DED8734E51EDDDB7F9; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 28 Apr 2010 19:39:03 GMT
Server:
```

```
\\{"alias":"sdavis","firstName":"sonya","lastName":"davis","dtmfAccessId":"12123","pkid":"c2e2bf1c-f249-40e5-b7b8-31a5b0333647"\\}
```

Import a User

```
POST http://<server>/vmrest/import/users/<type>?templateAlias=<voice mail user template>
with an ImportUser object as payload
```

An important point here is that because these users are imported, almost all values come from the internal database. One exception for an LDAP import is the dtmfAccessId. The dtmfAccessId data is not available in the internal database, so an LDAP import must specify both a pkid and a dtmfAccessId.

For example:

```
POST /vmrest/import/users/ldap?templateAlias=voicemailusertemplate HTTP/1.1
Content-Type: application/json
User-Agent: Java/1.6.0_20
Host: cuc-install-69.cisco.com
Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 126

\\{"alias":"sdavis","firstName":"sonya","lastName":"davis","dtmfAccessId":"12123","pkid":"c2e2bf1c-f249-40e5-b7b8-31a5b0333647"\\}
```

The POST above would produce the following response:

```
HTTP/1.1 201 Created
```