

Links to Other API pages: [Cisco Unity Connection APIs](#)

CUPI Guide Contents
API Overview
Index of All CUPI Documentation
Schedule Example

Contents

- [1 Overview of Cisco Unity Connection Schedule Objects](#)
 - ◆ [1.1 ScheduleSet Objects](#)
 - ◆ [1.2 Schedule Objects](#)
 - ◆ [1.3 ScheduleDetail Objects](#)
 - ◆ [1.4 ScheduleSetMemberMap Objects](#)
 - ◆ [1.5 Schedule Example](#)
- [2 CUPI for Schedules](#)
 - ◆ [2.1 ScheduleSets in CUPI](#)
 - ◇ [2.1.1 Retrieving the list of ScheduleSets](#)
 - [2.1.1.1 Listing Specific Tenant Related ScheduleSets by System Administrator](#)
 - ◇ [2.1.2 Retrieving a ScheduleSet](#)
 - ◇ [2.1.3 Adding a ScheduleSet](#)
 - ◇ [2.1.4 Changing a ScheduleSet](#)
 - ◇ [2.1.5 Deleting a ScheduleSet](#)
 - ◆ [2.2 Schedules in CUPI](#)
 - ◇ [2.2.1 Retrieving the list of Schedules](#)
 - [2.2.1.1 Listing Specific Tenant Related Schedules by System Administrator](#)
 - ◇ [2.2.2 Retrieving a Schedule](#)
 - ◇ [2.2.3 Adding a Schedule](#)
 - ◇ [2.2.4 Changing a Schedule](#)
 - ◇ [2.2.5 Deleting a Schedule](#)
 - ◆ [2.3 ScheduleSetMembers in CUPI](#)
 - ◇ [2.3.1 Retrieving the list of ScheduleSetMembers](#)
 - ◇ [2.3.2 Retrieving a ScheduleSetMember](#)
 - ◇ [2.3.3 Adding a ScheduleSetMember](#)
 - ◇ [2.3.4 Changing a ScheduleSetMember](#)
 - ◇ [2.3.5 Deleting a ScheduleSetMember](#)
 - ◆ [2.4 ScheduleDetails in CUPI](#)
 - ◇ [2.4.1 Retrieving the list of ScheduleDetails](#)
 - ◇ [2.4.2 Retrieving a ScheduleDetail](#)
 - ◇ [2.4.3 Adding a ScheduleDetail](#)
 - ◇ [2.4.4 Changing a ScheduleDetail](#)
 - ◇ [2.4.5 Deleting a ScheduleDetail](#)

Overview of Cisco Unity Connection Schedule Objects

Schedules in Cisco Unity Connection are somewhat complicated, since they are composed of several different types of objects. Before presenting the API for accessing schedules, it may be useful to review what these objects are and how they are used.

ScheduleSet Objects

ScheduleSet objects are the top-level objects. They are comprised of one or more Schedules, each of which is marked as included or excluded in the ScheduleSet.

Schedule Objects

Schedule objects are composed of one or more ScheduleDetail objects.

ScheduleDetail Objects

ScheduleDetail objects are the atomic schedule building blocks which comprise Schedule objects. They can be specified with Start and End Dates, Start and End Times of the day, and active Days of the Week. ScheduleDetails have a link to the Schedule objects that they are part of, and said Schedule object in a sense owns the ScheduleDetail object.

ScheduleSetMemberMap Objects

ScheduleSetMemberMap objects provide the linkage between a ScheduleSet and a Schedule that is included or excluded from it (via a boolean called Exclude). There will be one ScheduleSetMemberMap object for each Schedule that is included in or excluded from a ScheduleSet.

A ScheduleSetMemberMap abstracts the linkage between a ScheduleSet and a Schedule since neither object has an explicit linkage or ownership relationship of the other. One reason for this is that several ScheduleSets might reference the same Schedule (a Holiday Schedule for example). This differs from the relationship between a Schedule and a ScheduleDetail, since a ScheduleDetail has an explicit link to a Schedule, the Schedule essentially owns the ScheduleDetail, and no other Schedule may use another Schedule's ScheduleDetail object.

Schedule Example

For example, let's model a weekday schedule with the lunch hour blocked out as unavailable every work day, and with various holidays blocked out as well. Using the objects discussed previously, this schedule might be composed like so:

- First, we create a top-level ScheduleSet called WeekdaySet. WeekdaySet includes the WeekdaySchedule and excludes the HolidaySchedule (2 Schedule objects).
- After we create these 2 Schedule objects, we create 2 ScheduleSetMemberMap objects for WeekdaySet - one to include WeekdaySchedule and one to exclude HolidaySchedule.
- Then, we create 2 ScheduleDetail objects for WeekdaySchedule - one active from 8AM to 12PM Mon-Fri, and the other active from 1PM to 5PM Mon-Fri.

- Finally, we create various ScheduleDetails objects for HolidaySchedule, one per holiday. For example, we might create a July4ScheduleDetail with start and end dates both set to July 4 2010, and a WinterBreakScheduleDetail with a start date of Dec 23 2010 and an end date of Jan 3 2011.

To see how to create this schedule via CUPI, see the [Schedule Example page](#).

CUPI for Schedules

The previously described database objects are accessible to the administrator via CUPI. The following sections list the URIs to access these resources along with the data contained within them.

ScheduleSets in CUPI

ScheduleSets are top-level resources in CUPI, with a base URI of `+/vmrest/schedulesets+`.

A ScheduleSet object has the following fields:

Field Name	Field Type	Default	Notes
ObjectId	GUID	none	
TenantObjectId	Read Only	String (36)	The unique identifier of the tenant to which the schedulesets belong. This field is reflected in the response only if the schedulesets belong to a particular tenant.
DisplayName	String(64)	none	
OwnerLocattonObjectId	GUID	NULL	One of the Owners must be non-NULL
OwnerPersonalRuleSetObjectId	GUID	NULL	
OwnerSubscriberObjectId	GUID	NULL	
Undeletable	Boolean	False	Is only True for factory default objects

Retrieving the list of ScheduleSets

To retrieve the list of ScheduleSets, an administrator makes a GET to the schedulesets resource:

```
GET /vmrest/schedulesets
```

This would return the following on success:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<ScheduleSets total="3">
```

Cisco Unity Connection Provisioning Interface (CUPI)_API_--_Schedules

```
<ScheduleSet>
  <URI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da</URI>
  <ObjectId>30d9c0df-534b-437a-a6b7-439adfd850da</ObjectId>
  <DisplayName>Weekdays</DisplayName>
  <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>
  <OwnerLocationURI>/vmrest/locations/connectionlocations/6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationURI>
  <Undeletable>>true</Undeletable>
  <ScheduleSetMemberURI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmembers/30d9c0df-534b-437a-a6b7-439adfd850da</ScheduleSetMemberURI>
</ScheduleSet>
<ScheduleSet>
  <URI>/vmrest/schedulesets/75af01af-d290-4e0e-9862-5adf8293536c</URI>
  <ObjectId>75af01af-d290-4e0e-9862-5adf8293536c</ObjectId>
  <DisplayName>All Hours</DisplayName>
  <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>
  <OwnerLocationURI>/vmrest/locations/connectionlocations/6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationURI>
  <Undeletable>>true</Undeletable>
  <ScheduleSetMemberURI>/vmrest/schedulesets/75af01af-d290-4e0e-9862-5adf8293536c/schedulesetmembers/75af01af-d290-4e0e-9862-5adf8293536c</ScheduleSetMemberURI>
</ScheduleSet>
<ScheduleSet>
  <URI>/vmrest/schedulesets/e2e381e4-6096-4643-b0bb-b17a65b101bc</URI>
  <ObjectId>e2e381e4-6096-4643-b0bb-b17a65b101bc</ObjectId>
  <TenantObjectId>fe6541fb-b42c-44f2-8404-ded14cbf7438</TenantObjectId>
  <DisplayName>Voice Recognition Update Schedule</DisplayName>
  <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>
  <OwnerLocationURI>/vmrest/locations/connectionlocations/6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationURI>
  <Undeletable>>false</Undeletable>
  <ScheduleSetMemberURI>/vmrest/schedulesets/e2e381e4-6096-4643-b0bb-b17a65b101bc/schedulesetmembers/e2e381e4-6096-4643-b0bb-b17a65b101bc</ScheduleSetMemberURI>
</ScheduleSet>
</ScheduleSets>
```

Listing Specific Tenant Related ScheduleSets by System Administrator

In Cisco Unity Connection 10.5(2) and later, the system administrator can use TenantObjectId to list the specific tenant related schedulesets using the following URI:

```
GET https://<connection-server>/vmrest/schedulesets?query=(TenantObjectId is <Tenant-ObjectId>)
```

To get the TenantObjectId, use the following URI:

```
GET https://<connection-server>/vmrest/tenants
```

Retrieving a ScheduleSet

To retrieve a single ScheduleSet, an administrator makes a GET to the schedulesets resource, specifying the ObjectId of the requested ScheduleSet in the URI:

```
GET /vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da
```

This would return the following on success:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<ScheduleSet>
  <URI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da</URI>
  <ObjectId>30d9c0df-534b-437a-a6b7-439adfd850da</ObjectId>
  <DisplayName>Weekdays</DisplayName>
  <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>
  <OwnerLocationURI>/vmrest/locations/connectionlocations/6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationURI>
  <Undeletable>>true</Undeletable>
```

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_-_Schedules

```
<ScheduleSetMemberURI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmember/  
</ScheduleSet>
```

This would return the following if the specified ScheduleSet does not exist:

```
404  
Not Found  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<ErrorDetails>  
  <errors>  
    <code>NOT_FOUND</code>  
    <message>scheduleset - ObjectId=30d9c0df-534b-437a-a6b7-439adfd850da</message>  
  </errors>  
</ErrorDetails>
```

Adding a ScheduleSet

To add a new ScheduleSet, an administrator makes a POST to the schedulesets resource, specifying the new ScheduleSet via XML:

```
POST /vmrest/schedulesets  
  
<ScheduleSet>  
  <DisplayName>Night Shift</DisplayName>  
  <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>  
</ScheduleSet>
```

This will return the URI to the newly created ScheduleSet (which includes its ObjectId):

```
201  
Created  
/vmrest/user/schedulesets/0e58ec49-5064-4c9a-b1dc-dd47fe189419
```

Changing a ScheduleSet

To change an existing ScheduleSet, an administrator makes a PUT to the schedulesets resource, specifying the ObjectId of the ScheduleSet they wish to change in the URI and any data changes via XML:

```
PUT /vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da  
  
<ScheduleSet>  
  <DisplayName>Graveyard Shift</DisplayName>  
</ScheduleSet>
```

This would return the following on success:

```
204  
No Content
```

This would return the following if the specified scheduleset does not exist:

```
400  
Bad Request  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

Retrieving a ScheduleSet

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Schedules

```
<ErrorDetails>
  <errors>
    <code>DATA_EXCEPTION</code>
    <message>ScheduleSet not found </message>
  </errors>
</ErrorDetails>
```

Deleting a ScheduleSet

To delete an existing ScheduleSet, an administrator makes a DELETE to the schedulesets resource, specifying the ObjectId of the ScheduleSet they wish to delete in the URI:

```
DELETE /vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da
```

This would return the following on success:

```
204
No Content
```

This would return the following if the specified scheduleset does not exist:

```
404
Not Found
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ErrorDetails>
  <errors>
    <code>NOT_FOUND</code>
    <message>scheduleset - ObjectId=30d9c0df-534b-437a-a6b7-439adfd850da</message>
  </errors>
</ErrorDetails>
```

Note that if an administrator deletes a ScheduleSet object, then all of the supporting ScheduleSetMembers for that ScheduleSet will also be deleted.

Schedules in CUPI

Schedules are top-level resources in CUPI, with a base URI of +/vmrest/schedules+.

A Schedule object has the following fields:

Field Name	Field Type	Default	Notes
ObjectId	GUID	none	
TenantObjectId	Read Only	String (36)	The unique identifier of the tenant to which the schedule belong. This field is reflected in the response only if the schedule belong to a particular tenant.
DisplayName	String(64)	none	
OwnerLocationObjectId	GUID	NULL	One of the Owner's must be non-NULL
OwnerPersonalRuleSetObjectId	GUID	NULL	

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Schedules

OwnerSubscriberObjectId	GUID	NULL	
Undeletable	Boolean	False	Is only True for factory default objects
StartDate	DateTime	NULL	NULL means effective immediately
EndDate	DateTime	NULL	NULL means effective indefinitely
IsHoliday	Boolean	False	Holiday Greetings are used during this period if True

Retrieving the list of Schedules

To retrieve the list of Schedules, an administrator makes a GET to the schedules resource:

```
GET /vmrest/schedules
```

This would return the following on success:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<Schedules total="3">
  <Schedule>
    <URI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76</URI>
    <ObjectId>387f051e-3367-4cc8-abad-810293d39f76</ObjectId>
    <DisplayName>Weekdays</DisplayName>
    <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>
    <OwnerLocationURI>/vmrest/locations/connectionlocations/6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationURI>
    <Undeletable>true</Undeletable>
    <IsHoliday>false</IsHoliday>
    <ScheduleDetailsURI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails</ScheduleDetailsURI>
  </Schedule>
  <Schedule>
    <URI>/vmrest/schedules/75af01af-d290-4e0e-9862-5adf8293536c</URI>
    <ObjectId>75af01af-d290-4e0e-9862-5adf8293536c</ObjectId>
    <DisplayName>All Hours</DisplayName>
    <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>
    <OwnerLocationURI>/vmrest/locations/connectionlocations/6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationURI>
    <Undeletable>true</Undeletable>
    <IsHoliday>false</IsHoliday>
    <ScheduleDetailsURI>/vmrest/schedules/75af01af-d290-4e0e-9862-5adf8293536c/scheduledetails</ScheduleDetailsURI>
  </Schedule>
  <Schedule>
    <URI>/vmrest/schedules/e2e381e4-6096-4643-b0bb-b17a65b101bc</URI>
    <ObjectId>e2e381e4-6096-4643-b0bb-b17a65b101bc</ObjectId>
    <TenantObjectId>fe6541fb-b42c-44f2-8404-ded14cbf7438</TenantObjectId>
    <DisplayName>Holidays</DisplayName>
    <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>
    <OwnerLocationURI>/vmrest/locations/connectionlocations/6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationURI>
    <Undeletable>false</Undeletable>
    <IsHoliday>true</IsHoliday>
    <ScheduleDetailsURI>/vmrest/schedules/e2e381e4-6096-4643-b0bb-b17a65b101bc/scheduledetails</ScheduleDetailsURI>
  </Schedule>
</Schedules>
```

Cisco Unity Connection Provisioning Interface (CUPI) API -- Schedules

Listing Specific Tenant Related Schedules by System Administrator

In Cisco Unity Connection 10.5(2) and later, the system administrator can use TenantObjectID to list the specific tenant related schedules using the following URI:

```
GET https://<connection-server>/vmrest/schedules?query=(TenantObjectId is <Tenant-ObjectId>)
```

To get the TenantObjectID, use the following URI:

```
GET https://<connection-server>/vmrest/tenants
```

Retrieving a Schedule

To retrieve a single Schedule, an administrator makes a GET to the schedules resource, specifying the Objectid of the requested Schedule in the URI:

```
GET /vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76
```

This would return the following on success:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<Schedule>
  <URI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76</URI>
  <ObjectId>387f051e-3367-4cc8-abad-810293d39f76</ObjectId>
  <DisplayName>Weekdays</DisplayName>
  <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>
  <OwnerLocationURI>/vmrest/locations/connectionlocations/6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationURI>
  <Undeletable>true</Undeletable>
  <IsHoliday>false</IsHoliday>
  <ScheduleDetailsURI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails</ScheduleDetailsURI>
</Schedule>
```

This would return the following if the specified Schedule does not exist:

```
404
Not Found
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ErrorDetails>
  <errors>
    <code>NOT_FOUND</code>
    <message>schedule - ObjectId=387f051e-3367-4cc8-abad-810293d39f76</message>
  </errors>
</ErrorDetails>
```

Adding a Schedule

To add a new Schedule, an administrator makes a POST to the schedules resource, specifying the new Schedule via XML:

```
POST /vmrest/schedules

<Schedule>
  <DisplayName>EveningShift</DisplayName>
  <OwnerLocationObjectId>6a56503e-c1c8-406c-85fd-76be40994d39</OwnerLocationObjectId>
```


Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Schedules

```
<IsHoliday>>false</IsHoliday>
</Schedule>
```

This will return the URI to the newly created Schedule (which includes its ObjectID):

```
201
Created
/vmrest/user/schedules/df7faf3b-3278-4852-b488-7e3134dc3336
```

Changing a Schedule

To change an existing Schedule, an administrator makes a PUT to the schedules resource, specifying the ObjectID of the Schedule they wish to change in the URI and any data changes via XML:

```
PUT /vmrest/schedules/df7faf3b-3278-4852-b488-7e3134dc3336

<Schedule>
  <DisplayName>No Daylight Shift</DisplayName>
</Schedule>
```

This would return the following on success:

```
204
No Content
```

This would return the following if the specified schedule does not exist:

```
400
Bad Request
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ErrorDetails>
  <errors>
    <code>DATA_EXCEPTION</code>
    <message>Schedule not found </message>
  </errors>
</ErrorDetails>
```

Deleting a Schedule

To delete an existing Schedule, an administrator makes a DELETE to the schedules resource, specifying the ObjectID of the Schedule they wish to delete in the URI:

```
DELETE /vmrest/schedules/387f051e-3367-4cc8-abed-810293d39f76
```

This would return the following on success:

```
204
No Content
```

This would return the following if the specified schedule does not exist:

```
404
Not Found
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

Adding a Schedule

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Schedules

```
<ErrorDetails>
  <errors>
    <code>NOT_FOUND</code>
    <message>schedule - ObjectId=387f051e-3367-4cc8-abad-810293d39f76</message>
  </errors>
</ErrorDetails>
```

Note that if an administrator deletes a Schedule object, then all of the supporting ScheduleDetail objects for that Schedule will also be deleted.

ScheduleSetMembers in CUPI

ScheduleSetMembers are the resources representing ScheduleSetMemberMaps objects. They are sub-resources of ScheduleSets in CUPI, and thus are at a sub-URI of the schedulesets resource: `+/vmrest/scheduleset/{_}+<ScheduleSetObjectId>+{_}+/schedulesetmembers+`.

A ScheduleSetMember object has the following fields:

Field Name	Field Type	Default	Notes
ScheduleSetObjectId	GUID	none	Owning ScheduleSet
ScheduleObjectId	GUID	none	Schedule to include/exclude from ScheduleSet
Exclude	Boolean	False	False means Include, True means Exclude

Retrieving the list of ScheduleSetMembers

To retrieve the list of ScheduleSetMembers for a particular ScheduleSet, an administrator makes a GET to the schedulesetmembers resource, specifying the ObjectId of the ScheduleSet in the URI:

```
GET /vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmembers
```

This would return the following on success:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<ScheduleSetMembers total="2">
  <ScheduleSetMember>
    <URI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmembers/a1f34a57-b642-4d8b-9634-b4c2e37bfd2b</URI>
    <ScheduleSetObjectId>30d9c0df-534b-437a-a6b7-439adfd850da</ScheduleSetObjectId>
    <ScheduleSetURI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da</ScheduleSetURI>
    <ScheduleObjectId>a1f34a57-b642-4d8b-9634-b4c2e37bfd2b</ScheduleObjectId>
    <ScheduleURI>/vmrest/schedules/a1f34a57-b642-4d8b-9634-b4c2e37bfd2b</ScheduleURI>
    <Exclude>>false</Exclude>
  </ScheduleSetMember>
  <ScheduleSetMember>
    <URI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmembers/279fd73f-36a9-469a-9ad5-a0f80f09b2d</URI>
    <ScheduleSetObjectId>30d9c0df-534b-437a-a6b7-439adfd850da</ScheduleSetObjectId>
    <ScheduleSetURI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da</ScheduleSetURI>
    <ScheduleObjectId>279fd73f-36a9-469a-9ad5-a0f80f09b2d</ScheduleObjectId>
    <ScheduleURI>/vmrest/schedules/279fd73f-36a9-469a-9ad5-a0f80f09b2d</ScheduleURI>
    <Exclude>>true</Exclude>
  </ScheduleSetMember>
</ScheduleSetMembers>
```

Retrieving a ScheduleSetMember

To retrieve a single ScheduleSetMember for a particular ScheduleSet, an administrator makes a GET to the schedulesetmembers resource, specifying the ObjectIds of the ScheduleSet and the requested ScheduleSetMember in the URI:

```
GET /vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmembers/a1f34a57-b642-4d8
```

This would return the following on success:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<ScheduleSetMember>
  <URI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmembers/a1f34a57-b642-4d8b-9634-b4c2e37bfd2b</URI>
  <ScheduleSetObjectId>30d9c0df-534b-437a-a6b7-439adfd850da</ScheduleSetObjectId>
  <ScheduleSetURI>/vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da</ScheduleSetURI>
  <ScheduleObjectId>a1f34a57-b642-4d8b-9634-b4c2e37bfd2b</ScheduleObjectId>
  <ScheduleURI>/vmrest/schedules/a1f34a57-b642-4d8b-9634-b4c2e37bfd2b</ScheduleURI>
  <Exclude>>false</Exclude>
</ScheduleSetMember>
```

This would return the following if the specified ScheduleSetMember does not exist (meaning the Schedule is not listed as either included or excluded from the ScheduleSet):

```
404
Not Found
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ErrorDetails>
  <errors>
    <code>NOT_FOUND</code>
    <message>schedulesetmember - ObjectId=a1f34a57-b642-4d8b-9634-b4c2e37bfd2b</message>
  </errors>
</ErrorDetails>
```

Adding a ScheduleSetMember

To add a new ScheduleSetMember to a particular ScheduleSet, an administrator makes a POST to the schedulesetmembers resource, specifying the ObjectId of the ScheduleSet in the URI and the new ScheduleSetMember via XML:

```
POST /vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmembers
```

```
<ScheduleSetMember>
  <ScheduleSetObjectId>30d9c0df-534b-437a-a6b7-439adfd850da</ScheduleSetObjectId>
  <ScheduleObjectId>db46f878-bc72-4870-9482-9f1c336641ed</ScheduleObjectId>
  <Exclude>>false</Exclude>
</ScheduleSetMember>
```

This will return the URI to the newly created ScheduleSetMember (which includes its ObjectId):

```
201
Created
/vmrest/user/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmembers/68c53107-5d28-4f
```

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Schedules

NOTE: A ScheduleSet can have at most 1 Schedule marked as Included, and that Schedule must not be a Holiday Schedule. A ScheduleSet can also have at most 1 Schedule marked as Excluded, and that Schedule must be a Holiday Schedule. If an administrator makes a POST to the schedulesetmembers resource in an attempt to Include or Exclude more than 1 Schedule of a given type to a ScheduleSet, then CUPI will return an HTTP 400 error with a descriptive error message. Similarly, if an administrator makes a POST to the schedulesetmembers resource in an attempt to Include a Holiday Schedule or Exclude a non-Holiday Schedule, then CUPI will return an HTTP 400 error with a descriptive error message.

Changing a ScheduleSetMember

The schedulesetmembers resource does not support the PUT method. In order to change a ScheduleSetMember, an administrator must delete the existing one and then add it back with the requested change.

Deleting a ScheduleSetMember

To delete an existing ScheduleSetMember for a particular ScheduleSet, an administrator makes a DELETE to the schedulesetmembers resource, specifying the ObjectIds of the ScheduleSet and the ScheduleSetMember they wish to delete in the URI:

```
DELETE /vmrest/schedulesets/30d9c0df-534b-437a-a6b7-439adfd850da/schedulesetmembers/a1f34a57-b642-
```

This would return the following on success:

```
204
No Content
```

This would return the following if the specified schedulesetmember does not exist:

```
404
Not Found
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ErrorDetails>
  <errors>
    <code>NOT_FOUND</code>
    <message>schedulesetmember - ObjectId=a1f34a57-b642-4d8b-9634-b4c2e37bfd2b</message>
  </errors>
</ErrorDetails>
```

ScheduleDetails in CUPI

ScheduleDetails are sub-resources of Schedules in CUPI, and thus are at a sub-URI of the schedules resources: `+/vmrest/schedules/{_}<ScheduleObjectId>+{ _}/schedulesdetails+`.

A ScheduleDetail object has the following fields:

Field Name	Field Type	Default	Notes
------------	------------	---------	-------

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Schedules

ObjectId	GUID	none	
Subject	String(2048)	none	
ScheduleObjectId	GUID	none	Owning Schedule
StartDate	DateTime	NULL	NULL means effective immediately, also Time portion of DateTime is ignored
StartTime	Int	NULL	NULL means 12:00AM, otherwise minutes past 12:00AM so 480 means 8:00AM
EndDate	DateTime	NULL	NULL means effective indefinitely, also Time portion of DateTime is ignored
EndTime	Int	NULL	NULL means End-of-Day (11:59:59PM), otherwise minutes past 12:00AM so 1020 means 5:00PM
IsActiveMonday	Boolean	False	
IsActiveTuesday	Boolean	False	
IsActiveWednesday	Boolean	False	
IsActiveThursday	Boolean	False	
IsActiveFriday	Boolean	False	
IsActiveSaturday	Boolean	False	
IsActiveSunday	Boolean	False	

Retrieving the list of ScheduleDetails

To retrieve the list of ScheduleDetails for a particular Schedule, an administrator makes a GET to the scheduledetails resource, specifying the ObjectId of the Schedule in the URI:

```
GET /vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails
```

This would return the following on success:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<ScheduleDetails total="2">
  <ScheduleDetail>
    <URI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails/fb6cb280-ea91-4ee5-9225-6ca9c5e3b77e</URI>
    <ObjectId>fb6cb280-ea91-4ee5-9225-6ca9c5e3b77e</ObjectId>
    <ScheduleObjectId>387f051e-3367-4cc8-abad-810293d39f76</ScheduleObjectId>
    <ScheduleURI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76</ScheduleURI>
    <Subject>Weekday Mornings</Subject>
    <StartTime>480</StartTime>
    <EndTime>720</EndTime>
    <IsActiveMonday>true</IsActiveMonday>
    <IsActiveTuesday>true</IsActiveTuesday>
    <IsActiveWednesday>true</IsActiveWednesday>
    <IsActiveThursday>true</IsActiveThursday>
    <IsActiveFriday>true</IsActiveFriday>
    <IsActiveSaturday>false</IsActiveSaturday>
```

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Schedules

```
<IsActiveSunday>>false</IsActiveSunday>
</ScheduleDetail>
<ScheduleDetail>
  <URI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails/e049dc1e-7447-4e2a-907e-03d67e2d40a1</URI>
  <ObjectId>e049dc1e-7447-4e2a-907e-03d67e2d40a1</ObjectId>
  <ScheduleObjectId>387f051e-3367-4cc8-abad-810293d39f76</ScheduleObjectId>
  <ScheduleURI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76</ScheduleURI>
  <Subject>Weekday Afternoons</Subject>
  <StartTime>780</StartTime>
  <EndTime>1020</EndTime>
  <IsActiveMonday>>true</IsActiveMonday>
  <IsActiveTuesday>>true</IsActiveTuesday>
  <IsActiveWednesday>>true</IsActiveWednesday>
  <IsActiveThursday>>true</IsActiveThursday>
  <IsActiveFriday>>true</IsActiveFriday>
  <IsActiveSaturday>>false</IsActiveSaturday>
  <IsActiveSunday>>false</IsActiveSunday>
</ScheduleDetail>
</ScheduleDetails>
```

Retrieving a ScheduleDetail

To retrieve a single ScheduleDetail for a particular Schedule, an administrator makes a GET to the scheduledetails resource, specifying the ObjectIds of the Schedule and the requested ScheduleDetail in the URI:

```
GET /vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails/fb6cb280-ea91-4ee5-9225-6ca9c5e3b77e
```

This would return the following on success:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<ScheduleDetail>
  <URI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails/fb6cb280-ea91-4ee5-9225-6ca9c5e3b77e</URI>
  <ObjectId>fb6cb280-ea91-4ee5-9225-6ca9c5e3b77e</ObjectId>
  <ScheduleObjectId>387f051e-3367-4cc8-abad-810293d39f76</ScheduleObjectId>
  <ScheduleURI>/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76</ScheduleURI>
  <Subject>Weekday Mornings</Subject>
  <StartTime>480</StartTime>
  <EndTime>720</EndTime>
  <IsActiveMonday>>true</IsActiveMonday>
  <IsActiveTuesday>>true</IsActiveTuesday>
  <IsActiveWednesday>>true</IsActiveWednesday>
  <IsActiveThursday>>true</IsActiveThursday>
  <IsActiveFriday>>true</IsActiveFriday>
  <IsActiveSaturday>>false</IsActiveSaturday>
  <IsActiveSunday>>false</IsActiveSunday>
</ScheduleDetail>
```

This would return the following if the specified ScheduleDetail does not exist:

```
404
Not Found
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ErrorDetails>
  <errors>
    <code>NOT_FOUND</code>
    <message>scheduledetail - ObjectId=fb6cb280-ea91-4ee5-9225-6ca9c5e3b77e</message>
  </errors>
</ErrorDetails>
```

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Schedules

```
</errors>  
</ErrorDetails>
```

Adding a ScheduleDetail

To add a new ScheduleDetail to a particular Schedule, an administrator makes a POST to the scheduledetails resource, specifying the ObjectId of the Schedule in the URI and the new ScheduleDetail via XML:

```
POST /vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails
```

```
<ScheduleDetail>  
  <Subject>Saturday (Half-day)</Subject>  
  <StartTime>540</StartTime>  
  <EndTime>780</EndTime>  
  <IsActiveSaturday>true</IsActiveSaturday>  
</ScheduleDetail>
```

This will return the URI to the newly created ScheduleDetail (which includes its ObjectId):

```
201  
Created  
/vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails/9d8afb61-bd4b-4e65-b274-e87
```

Changing a ScheduleDetail

To change an existing ScheduleDetail for a particular Schedule, an administrator makes a PUT to the scheduledetails resource, specifying the ObjectId of the Schedule and the ScheduleDetail they wish to change in the URI and the data changes via XML:

```
PUT /vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails/fb6cb280-ea91-4ee5-9225
```

```
<ScheduleDetail>  
  <StartTime>450</StartTime>  
</ScheduleDetail>
```

This would return the following on success:

```
204  
No Content
```

This would return the following if the specified scheduledetail does not exist:

```
400  
Bad Request  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<ErrorDetails>  
  <errors>  
    <code>DATA_EXCEPTION</code>  
    <message>ScheduleDetail not found </message>  
  </errors>  
</ErrorDetails>
```

Deleting a ScheduleDetail

To delete an existing ScheduleDetail from a particular Schedule, an administrator makes a DELETE to the scheduledetails resource, specifying the ObjectId of the Schedule and the ScheduleDetail they wish to delete in the URI:

```
DELETE /vmrest/schedules/387f051e-3367-4cc8-abad-810293d39f76/scheduledetails/9d8afb61-bd4b-4e65-b
```

This would return the following on success:

```
204  
No Content
```

This would return the following if the specified scheduledetail does not exist:

```
404  
Not Found  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<ErrorDetails>  
  <errors>  
    <code>NOT_FOUND</code>  
    <message>scheduledetail - ObjectId=9d8afb61-bd4b-4e65-b274-e870a2b51865</message>  
  </errors>  
</ErrorDetails>
```