Web Services Reference for Cisco Unified Contact Center Domain Manager

Release 11.0.1

August 2015

Corporate Headquarters
Cisco Systems, Inc.
170, West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Table of Contents

Preface

Purpose

This document describes the Web Service APIs that Unified CCDM exposes and explains how to use them. The Web Services APIs allow resource and hierarchy management operations to be performed remotely from a third party client application. The Web Services APIs also allow the client to subscribe to notifications about state changes in contact center resources.

Audience

This document is intended for web developers who are writing applications that need to use the Unified CCDM Web Service APIs. The reader should be familiar with web service integration, and also have a basic understanding of Unified CCDM and how it interacts with the contact center environment.

Organization

This document contains the following chapters:

Chapter 1	Getting Started	Provides basic information on the Unified CCDM environment and how to use the context-sensitive online help.
Chapter 2	Settings	Describes how users can change their settings, or the settings of others, such as email address, time-zone, or security privileges.
Chapter 3	Service Manager	Explains how supervisor users can use Unified CCDM for management of agents, teams, and skill groups.
Chapter 4	Information Notices	Explains how to view and edit information notices.

Chapter 5	System Manager	Describes the folder structure of Unified CCDM and explains how to set up and maintain agents, contacts, equipment, and other resources using Unified CCDM.
Chapter 6	System Manager Resource Guide	Provides detailed information about each provisionable resource type.
Chapter 7	Security Manager	Explains how to set up security settings for individual users by means of roles and tasks, and how users with the same permissions can be added to groups to simplify management.
Chapter 8	System Administration Tasks	Explains the main Unified CCDM administrative tasks.

Document Conventions

This document uses the following conventions:

Convention	Description
boldface font	Boldface font is used to indicate commands, such as entries, keys, buttons, folders and submenu names. For example: • Choose **Edit > Find** • Click **Finish**
italic font	Italic font is used to indicate the following: • To introduce a new term; for example: A *skill group* is a collection of agents who share similar skills • For emphasis; for example: *Do not* use the numerical naming convention • A syntax value that the user must replace; for example: IF (*condition, true-value, false-value*) • A title of a publication; for example: Refer to the *Cisco CRS Installation Guide*
`window font`	Window font, such as Courier, is used for the following: • Text as it appears in code or that the window displays; for example: `<html><title>Cisco Systems, Inc. </title></html>`
< >	Angle brackets are used to indicate the following: • For arguments where the context does not allow italic, such as ASCII output • A character string that the user enters but that does not appear on the window, such as a password

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation, at:

http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

Subscribe to What's New in Cisco Product Documentation, which lists all new and revised Cisco technical documentation as an RSS feed and delivers content directly to your desktop using a reader application. The RSS feeds are a free service.

Field Alerts and Field Notices

Cisco products may be modified or key processes may be determined to be important. These are announced through use of the Cisco Field Alerts and Cisco Field Notices. You can register to receive Field Alerts and Field Notices through the Product Alert Tool on Cisco.com. This tool enables you to create a profile to receive announcements by selecting all products of interest.

Log into www.cisco.com and then access the tool at http://www.cisco.com/cisco/support/notifications.html

Documentation Feedback

To provide comments about this document, send an email message to the following address: contactcenterproducts_docfeedback@cisco.com

We appreciate your comments.

1 What's Changed in this Release

1.1 CCDM 10.6

The following items have changed between CCDM 10.0 and CCDM 10.6.

1.1.1 Additional Resources Supported

The Resource Management web service now supports the following additional resources:

- IVR app
- Media file

1.1.2 Other Changes

- The following fields have been removed from the Resource Management `Person` item:
  - `MiddleName`
  - `Title`
- The `Errors` field in the Resource Management `RequestResult` item has been clarified.
- The `Identity` field in the Resource Management `Entry` item has been clarified.
- The `LoginName` field in the Resource Management `User` item has been clarified for ISE-enabled and Single sign-on users.
- There is a new field, `HomeFolder`, for the Resource Management `User` item.
- The `Item` field in the Resource Management `Dialed Number Call Type Member` is now mandatory.
- The valid state translations for the Resource Management `Status` field have been updated.
- Descriptions of the following missing field in Resource Management is now included:
  - `User` item, `NewPassPhrase` field.

1.1.3 Additional Resources Supported

The Resource Management web service now supports the following additional resources and memberships:

- Category
- Category Item membership

1.1.4 Associated Name Fields for Sorting Reach Results

Many resource items have associated (parent) name fields linked to some of the resource properties. An associated field can be specified as a sort term for the `Search()` method instead of the corresponding resource property . This allows you to order the search results alphabetically by name, rather than by a system id value.

For example, when searching for Agents, you can now specify a sort term of

```
sort:ParentPeripheralDisplayName,ASC
```

to return the agents that match the search criteria, sorted by peripheral name. Previously you would have needed to sort by `PeripheralUrn`, and include a separate step to convert the returned results into a more useful order.

1.1.5 Other Changes

- The type returned by the `Audit()` method has changed.
- The following fields have been removed from `DimensionItem`:
  - `CurrencyUrn`
  - `GeographyUrn`
  - `PrimaryLanguageUrn`
  - `SecondaryLanguageUrn`
  - `TertiaryLanguageUrn`.

1.2 CCDM 10.0

The following items have changed between CCDM 9.2 and CCDM 10.0.

1.2.1 Item Ids No Longer Globally Unique

Item ids are no longer globally unique, although they are still unique for an item type. The following changes support this:

- The Resource Management `Search()` method may not return a single item for the search term item:<Id>. To locate a specific item using an item id it is now necessary to specify the type: term as well. Otherwise the search will return all items of all types with the specified id.

1.2.2 REST Support

The Resource Management web service now supports the REST protocol.

1.2.3 New Resource Management Methods

The following Resource Management methods have been added:

- `Audit()` returns the audit records associated with one or more resource entities.
- `Save()` to create, update and delete multiple item and types in a single call.
- `Move()` moves one or more resource items to the specified folder from their current folder location or locations.
- `Upload()` allows binary data to be associated with an existing resource and uploaded to the appropriate remote system (currently supported for media files only).
- `Download()` returns the binary data that has been associated with a resource (currently supported for media files only).
- `Deploy()` allows bulk update of one or more resources of a single type from a single binary deployment package (currently supported for IVR scripts only).

1.2.4 Changed Resource Management Error Type

The Resource Management Web services now return a `ResourceFaultDetail` item (see section section 4.3.4.6 "Resource Fault Detail") when a method fails. Previously an `ExceptionDetail` item was returned.

1.2.5 Support for Mapping Resources to Multiple Equipment Instances

This release includes support for pkey maps, which can be used link resources to multiple remote equipment instances with different details on each. If the client is not interested in linking resources to multiple remote equipment instances, then the pkey map implementation is almost transparent (except for changes in the results returned by the `Describe()` method).

The Entry class has a new `EquipmentMapping` field where pkey map information is stored, and there are new `PkeyMap`, `ItemPkey` and classes.

The following new provisionable pkey map item types and pkey map member types have been defined:

- Agent Pkey
- Agent Desktop Pkey

- Agent Team Pkey
- Call Type Pkey
- Calling Search Space Pkey
- Dialed Number Pkey
- Enterprise Skill Group Pkey
- Expanded Call Variable Pkey
- Label Pkey
- Network VRU Script Pkey
- Person Pkey
- Precision Attribute Pkey
- Precision Queue Pkey
- Precision Queue Step Pkey
- Route Pkey
- Route Partition Pkey
- Service Pkey
- Skill Group Pkey
- Tenant Pkey
- User Variable Pkey
- Agent Agent Team Member Pkey
- Agent Peripheral Member Pkey
- Agent Precision Attribute Member Pkey
- Agent Skill Group Member Pkey
- Call Type Routing Script Member Pkey
- Dialed Number Call Type Member Pkey
- Precision Queue Step Precision Attribute Member Pkey
- Query Rule Campaign Member Pkey
- Route Partition Calling Search Space Member Pkey
- Skill Group Campaign Member Pkey
- Skill Group Service Member Pkey

1.2.6 Support for Precision Queues

The Resource Management web service includes support for Precision Queues. The following new provisionable item types and member types have been defined:

- Precision Queues
- Precision Queue Steps
- Precision Attributes.
- Agent Precision Attribute Member,
- Precision Queue Bucket Interval Member,
- Precision Queue Step Precision Attribute Member
- Precision Queue Step Precision Queue Member

Note that a precision queue is a composite type, and must be created in a batch that includes at least one precision queue step and one attribute member.

1.2.7 Additional Resources Supported

The Resource Management web service now supports the following additional resources and memberships:

- Agent desktop
- Calling search space
- Enterprise skillgroup
- Expanded call variable
- IVR script (or VXML application)
- Label
- Media file
- Network VRU script
- Route
- Route partition
- Service
- User variable
- Department

1.2.8 Resource Management WSDL Changes for New Methods

The Resource Management web service WSDL has been updated to include the new methods.

1.2.9 Resource Management WSDL Changes for PKey Maps

The Resource Management web service WSDL has been updated to include the new pkey map functionality.

1.2.10 New and Changed Fields

The following Resource Management web service have been added:

- The Skillgroup resource item has a new field, `IsPRSkill`, that indicates whether the Skillgroup is a precision routing skill group.

The following Resource Management web service field names have changed:

- In the Dialed Number resource item, the `Digits` field has been renamed to `DialedNumber`.

1.2.11 Other Changes

- The `DirectID` search term has been removed
- The Resource Management classes have been restructured to use abstract classes for common fields.
- New system types, `IT_CLUSTER_CONNECTION_*`and `IT_CLUSTER_RESOURCE_*`, have been added to allow you query the cluster configuration.
- The descriptions of the search terms used by the Resource Management `Search()` method have been enhanced and clarified.
- The following search types have been reinstated for backwards compatibility: `member`, `memberchildof` and `memberparentof`.
- Information has been added explaining how to use SSL to secure communication with the notifications subscriber server.
- Additional information has been added about type-2 SCDs.

2 Getting Started

2.1 Unified CCDM Web Services

2.1.1 About Unified CCDM Web Services

Unified CCDM exposes Web Services APIs that can be called by third party client applications.

This document describes the Web Services APIs that Unified CCDM exposes and how to use them.

Table 2.1 Unified CCDM Web Services

Web Service	Description
Resource Management	Allows the client to provision and manage contact center resources (see Chapter 4 "Resource Management Web Service").
Subscriptions	Allows the client to subscribe to notifications about state changes to contact center resources (see Chapter 5 "Subscriptions Web Service").

For a technical overview of Unified CCDM, including architecture, design concepts and performance tips, see section 5.6.3 "Dialed Number From Unallocated folder".

2.1.2 About Web Service Protocols

The Unified CCDM Web Services support two protocols:

- Simple Object Access Protocol, or SOAP (see http://en.wikipedia.org/wiki/SOAP)
- Representational state transfer, or REST (see http://en.wikipedia.org/wiki/REST)

Typically, inter-platform interactions, such as an Enterprise Service Bus, use SOAP because of its excellent standards-based interoperability, ease of multi-firewall traversal and existing engineering skill sets, whereas thin client and mobile applications typically use REST because of its low footprint and ease of coding.

All the Unified CCDM Web Services support SOAP as a default protocol. In addition, the Resource Management Web Service also supports the REST protocol

with the same payload types (using both XML and JSON content types). The SOAP implementation is also built on architectural REST principles but within the SOAP envelope.

This simultaneous support of both SOAP and REST allows for maximum interoperability and extends the traditional Unified CCDM API use to client gadgets.

The following table shows the typical usages of the Unified CCDM Web Services.

Table 2.2 Typical Unified CCDM Web Services Usage

Web Service	SOAP	REST XML	REST JSON
Resource Management	Enterprise Service Bus / Thick client	Java or other thin client	JavaScript gadgets/widgets and mobile
Subscriptions	Enterprise Service Bus/ Thick client	N/A, SOAP only concept	N/A, SOAP only concept

2.1.2.1 SOAP

The Unified CCDM SOAP web service design concepts are:

- The technology base is built on a SOAP 1.2 web service stack to address the widest market at present.
- For all Unified CCDM Web Services, the WS-Security and WS-Addressing standards are implemented.
- The WS-Eventing and Subscription services have been implemented to allow for asynchronous notification of changes of resources.
- The Unified CCDM Resource Management Web Service has combined elements of WS-Resourcing and WS-Management (these could not be used directly since both contain proprietary elements).

2.1.2.2 REST

The provisioning resources are organized into collections to allow clients and the Unified CCDM server to perform queries on the collections and to use the collection as a factory for creating new resources. Since they are collection resources they are plural nouns with a URI path that indicates the hierarchy. Unified CCDM supports three types of resource addressing:

1. Classic resource addressing. This is the usual REST collection resource paradigm where one or more resources of the same type may be contained in a path. The URI is hierarchical and contains:
    - the resource type as part of the path

- a single identifier to get, put or delete a resource of that type.

For example

```
GET /resources/agents/1006
```

where `1006` is the entity id and `agents` is the type of resource.

2. Composite resource collection. This is a collection URI where the identifier is a composite key made up of the entity id and the entity type. Resources of all types may be addressed under a common resources path. This is typically used where provisioning resources of all types may be held.

   For example

   ```
   GET /resources/1006,Agent
   ```

3. Anonymous collections. These are an extension of both classic and composite resource addressing and use multiple ids to provide bulk operation capability. The key to understanding this extension is that a sub-collection of resources is itself a collection which can be cached. The sub-collection identifier is the concatenation of entity ids (and possible types). Both classic and composite addressing modes are supported:

   For example

   ```
   GET /resources/agents/1006,1007,1008GET
   /resources/1006,Agent|3412,Label
   ```

Table 2.3 Supported Addressing Modes

	Create	Update	Describe / Search	Retrieve	Delete	Audit
REST Verb	POST	PUT	GET	GET	DELETE	GET
Collection Type:						
Classic Addressing resource/{type}/{id}	Yes	Yes	No	Yes	Yes	Yes
Composite Addressing resource/{id,type}	Yes	Yes	No	Yes	Yes	No
Batch Classic Addressing resource/{type}/{id},{id},{id},…	Yes	Yes	No	Yes	Yes	No
Batch Composite Addressing resource/{id,type}|{id,type}…	Yes	Yes	No	Yes	Yes	No

2.2 Security

2.2.1 Resource Management and Subscriptions Web Services

The Unified CCDM Web Services use Unified CCDM's role-based access control system to secure and restrict what the client can do or see. The authentication mode depends on the protocol being used.

The Unified CCDM Web Services are secured using WS-Security username tokens. You must pass a valid Unified CCDM username and password to access the service methods.

2.2.1.1 SOAP

Tip
You can use the SoapUI tool, Project Settings window, Security Configuration tab to configure a security profile. This profile can then be selected for each request using the pop-up Auth tab on the request editor window.

Figure 2.1 "Configuring a Security profile using SOAP UI" shows how to configure a security profile using SOAP UI.

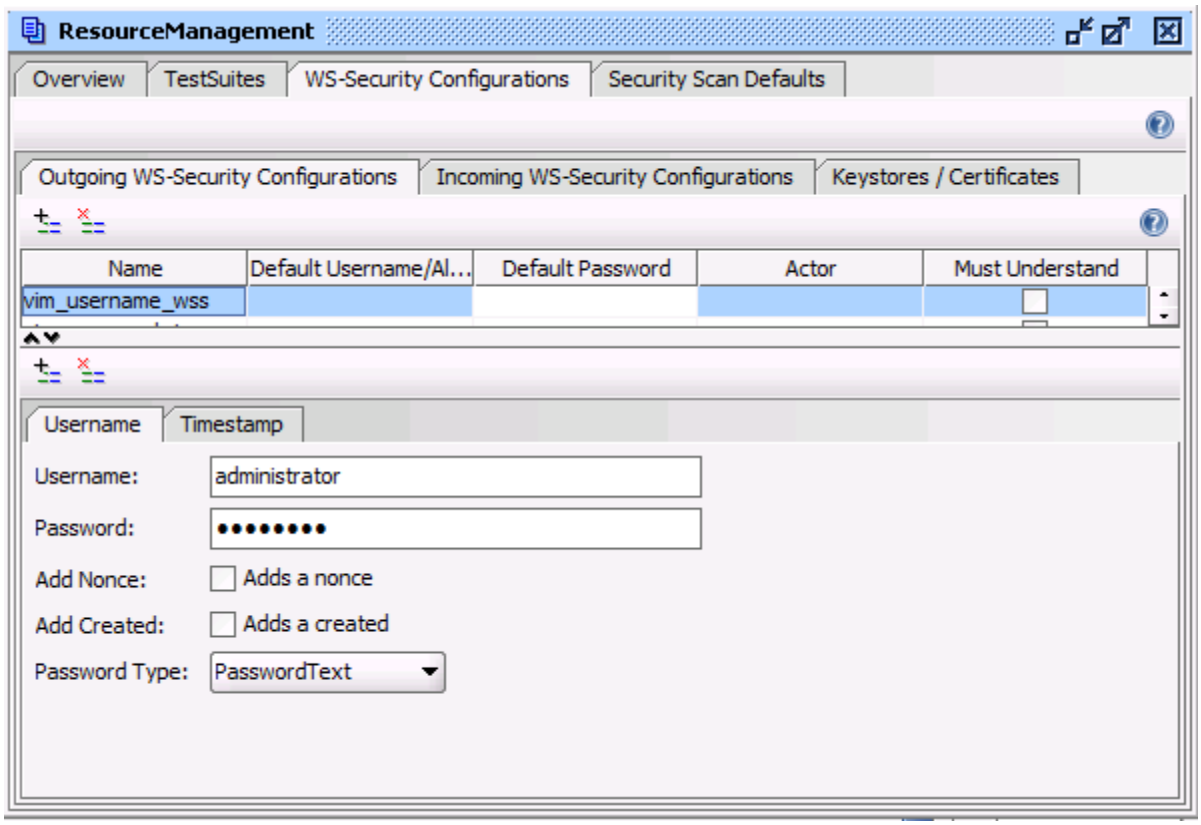


Figure 2.1 Configuring a Security profile using SOAP UI

Figure 2.2 "A WS-Security section in a SOAP Header" shows how to add a WS-Security section to the SOAP header sent to the Unified CCDM Server.

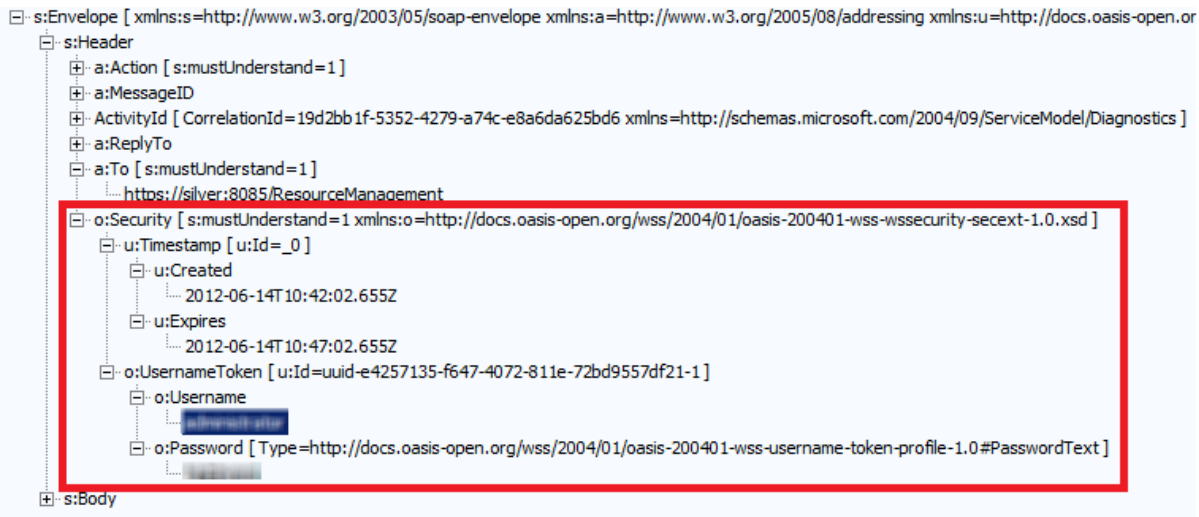


Figure 2.2 A WS-Security section in a SOAP Header

2.2.1.2 REST

The Resource Management Web Service uses Basic Authentication with the REST protocol. In this case, the client encodes the user name and password in Base-64 and adds it to the HTTP header.

An example is

```
GET /private/index.html HTTP/1.1
Host: local host
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

Note

The implementation does not support challenge and response and so the responses cannot be viewed using a browser.

2.2.2 Notifications

Notifications sent to clients who have subscribed for them using the Subscriptions Web Service are secured using mutual certificates.

For more information about configuring security for notifications, see section 5.3 "Securing Notifications with SSL".

3 Common Data Types

3.1 Remote Resource Types

3.1.1 About Remote Resource Types

These types correspond directly to the remote resources in the Contact Center environment in which Unified CCDM operates. These types can be used with the Resource Management Web Service APIs (see Chapter 4 "Resource Management Web Service") to allow client utilities and a mid-tier interface to perform adds, updates and deletes.

3.1.2 Provisionable Remote Resource Types

These remote resource types are fully supported by the Resource Management Web Service.

Resource Type	Internal Name	REST Parameter
Agent	IT_AGENT	agent
Agent Desktop	IT_AGENT_DESKTOP	agent-desktop
Agent Team	IT_AGENT_TEAM	agent-team
Call Type	IT_CALL_TYPE	call-type
Calling Search Space	IT_CALLING_SEARCH_SPACE	calling-search-space
Department	IT_DEPARTMENT	department
Dialed Number	IT_DIALED_NUMBER	dialed-number
Enterprise Skill Group	IT_ENTERPRISE_SKILLGROUP	enterprise-skill-group
Expanded Call Variable	IT_EXPANDED_CALL_VARIABLE	expanded-call-variable
IVR Script (or VXML Application)	IT_IVR_SCRIPT	ivr-script
Label	IT_LABEL	label
Media file	IT_MEDIAFILE	mediafile
Network VRU Script	IT_NETWORK_VRU_SCRIPT	network-vru-script
Person	IT_PERSON	person

Resource Type	Internal Name	REST Parameter
Precision Attribute	IT_PRECISION_ATTRIBUTE	precision-attribute
Precision Queue	IT_PRECISION_QUEUE	precision-queue
Precision Queue Step	IT_PRECISION_QUEUE_STEP	precision-queue-step
Route	IT_ROUTE	route
Route Partition	IT_ROUTE_PARTITION	route-partition
Service	IT_SERVICE	service
Skill Group	IT_SKILLGROUP	skill-group
Tenant	IT_TENANT	tenant
User Variable	IT_USER_VARIABLE	user-variable

3.1.3 Non Provisionable Remote Resource Types

These remote resource types are supported by the Resource Management Web Services for searching only.

Resource Type	Internal Name
Announcement	IT_ANNOUNCEMENT
Application Gateway	IT_APPLICATION_GATEWAY
Application Instance	IT_APPLICATION_INSTANCE
Bucket Interval	IT_BUCKET_INTERVAL
Call Source	IT_CALL_SOURCE
Campaign	IT_CAMPAIGN
Chargeband	IT_CHARGEBAND
Cli	IT_CLI
Custom Entry	IT_CUSTOM_ENTRY
Date Time Setting	IT_DATE_TIME_SETTING
Device Target	IT_DEVICE_TARGET
Dialer	IT_DIALER
Dial Number Plan	IT_DIAL_NUMBER_PLAN

Resource Type	Internal Name
Enterprise Route	IT_ENTERPRISE_ROUTE
Enterprise Service	IT_ENTERPRISE_SERVICE
Gateway Function	IT_GATEWAY_FUNCTION
Gateway Result	IT_GATEWAY_RESULT
Gateway Server	IT_GATEWAY_SERVER
ICR Instance	IT_ICR_INSTANCE
Import Rule	IT_IMPORT_RULE
IVR Entry Point	IT_IVR_ENTRY_POINT
IVR Module	IT_IVR_MODULE
IVR Routing Target	IT_IVR_ROUTING_TARGET
IVR Script Node	IT_IVR_SCRIPT_NODE
Logical Interface Controller	IT_LOGICAL_INTERFACE_ CONTROLLER
Media Class	IT_MEDIA_CLASS
Media File Server	IT_MEDIAFILE_SERVER
Media Routing Domain	IT_MEDIA_ROUTING_DOMAIN
Network Trunk Group	IT_NETWORK_TRUNK_GROUP
Network Vru	IT_NETWORK_VRU
Object Type	IT_OBJECT_TYPE
Peripheral	IT_PERIPHERAL
Physical Interface Controller	IT_PHYSICAL_INTERFACE_ CONTROLLER
Port	IT_PORT
Query Rule	IT_QUERY_RULE
Rating Period	IT_RATING_PERIOD
Reason Code	IT_REASON_CODE
Region	IT_REGION

Resource Type	Internal Name
Route Partition	IT_ROUTE_PARTITION
Routing Client	IT_ROUTING_CLIENT
Routing Script	IT_ROUTING_SCRIPT
Schedule	IT_SCHEDULE
Scheduled Target	IT_SCHEDULED_TARGET
Script	IT_SCRIPT
Script Node	IT_SCRIPT_NODE
Strategy	IT_STRATEGY
Timeband	IT_TIMEBAND
Tli	IT_TLI
Trunk	IT_TRUNK
Trunk Group	IT_TRUNK_GROUP
VXML Application Server (sometimes called IVR Script Server)	IT_VXMLAPPLICATION_SERVER
Wrapup Code	IT_WRAPUP_CODE

3.2 System Resource Types

3.2.1 About System Resource Types

System Resource types are contained in the Unified CCDM database. They are not provisioned to remote contact center equipment, but are Unified CCDM-specific items used for organizing and accessing other resource types.

3.2.2 Editable System Resource Types

These system resource types are fully supported by the Resource Management Web Services.

Resource Type	Internal Name	REST Parameter
Folder	IT_FOLDER	folder
Group	IT_GROUP	group
User	IT_USER	user

3.2.3 Read-Only System Resource Types

These system resource types are supported by the Resource Management Web Services for searching only.

Resource Type	Internal Name
Date	IT_DATE
NNG	IT_NNG
Time	IT_TIME
Cluster Connection	IT_CLUSTER_CONNECTION
Cluster Connection Type	IT_CLUSTER_CONNECTION_TYPE
Cluster Resource	IT_CLUSTER_RESOURCE
Cluster Resource Instance	IT_CLUSTER_RESOURCE_INSTANCE
Cluster Resource Instance Component	IT_CLUSTER_RESOURCE_INSTANCE_COMPONENT
Cluster Resource Instance Component Type	IT_CLUSTER_RESOURCE_INSTANCE_COMPONENT_TYPE
Cluster Resource Type	IT_CLUSTER_RESOURCE_TYPE

Resource Type	Internal Name
Cluster Resource Type Group	IT_CLUSTER_RESOURCE_TYPE_GROUP
Cluster Server	IT_CLUSTER_SERVER
Custom 0	IT_CUSTOM0
Custom 1	IT_CUSTOM1
Custom 2	IT_CUSTOM2
Custom 3	IT_CUSTOM3
Custom 4	IT_CUSTOM4
Custom 5	IT_CUSTOM5
Custom 6	IT_CUSTOM6
Custom 7	IT_CUSTOM7
Custom 8	IT_CUSTOM8
Custom 9	IT_CUSTOM9
Fileshare	IT_FILESHARE
Email	IT_EMAIL
Form Template	IT_FORM_TEMPLATE
Hierarchy	IT_HIERARCHY
Hierarchy Node	IT_HIERARCHY_NODE
Managerial Role	IT_MANAGERIAL_ROLE
Organisation	IT_ORGANISATION
Printer	IT_PRINTER
Product	IT_PRODUCT
Question	IT_QUESTION
Speciality	IT_SPECIALITY

3.3 About Resources and SCD Effective Dating

Remote resource types and their memberships are modeled as type-2 Slowly Changing Dimension (SCD) properties. The full resource history is maintained by creating new time-bounded records for each significant change (see http://en.wikipedia.org/wiki/Slowly_changing_dimension for more information).

- The Resource Management Web Service uses SCDs for memberships only, but not items. For instance, re-skilling an agent from one skill group to another will result in an SCD change for the agent-skill group member whereas changing the details of the agent or the skill group will not.

Tip
Item effective dating can be used to schedule a provisioning operation to occur in the future, that is, setting the `EffectiveFrom` field to be a future date.

The resource fields used to maintain SCD properties are `EffectiveFrom`, `EffectiveTo` and `Latest`. Together, these fields define the life cycle of the resource items and memberships.

For example, a remote resource may have the following values.

Identifier	Surrogate	EffectiveFrom	EffectiveTo	Latest
1000	1000	01/01/2012	06/06/2079	1

After a SCD event the resource may have the following values.

Identifier	Surrogate	EffectiveFrom	EffectiveTo	Latest
1000	1000	01/01/2012	02/02/2012	0
1000	1001	02/02/2012	06/06/2079	1

The surrogate key field is used to track the change.

4 Resource Management Web Service

4.1 About the Resource Management Web Service

4.1.1 Description

The Resource Management Web Service provides APIs that allow third party client applications to invoke provisioning operations on the underlying equipment and to create system resources.

4.1.2 Concepts

The Resource Management Web Service APIs have the following high level concepts:

- Provisioning operations are based on:
  - *Resources*, such as Agents and Call Types
  - *Resource Memberships* which define the relationship between these Resources.
- Most resources memberships are many-to-many associations. This model is loosely based on the Frameworx Shared Information/Data model (SID), with extensions to support full multi-tenancy and type-2 SCD life-cycle management. See http://www.tmforum.org/TMForumFrameworx/1911/home.html for more information about the Frameworx SID model.
- Resources are organized in a folder tree, where each top-level folder is either owned by a tenant or is one of the special folders (for example **/Unallocated**, **/Shared** or **/Search**).
- Each resource is dated with a common header which uniquely identifies it so its life-cycle can be tracked. Type-2 SCD life-cycle tracking is used, so there is the concept of a latest resource and previous resources, each of which has its own identifier.
- Each resource or resource member has a common header which uniquely identifies the resource, and a loosely-coupled data structure body. This structure has two advantages:
  - New resource types can be added without needing to regenerate the WSDL (SOAP) or to change existing clients (REST).
  - Versioning can be performed at the resource or resource member level and not at the API level, which reduces upgrade costs.

- Transactions are coarse grained, and each create, update, save, move or retrieve web service call supports an array of up to 100 operations to minimize API chatter.
- Provisioning transactions may be synchronous or asynchronous:
  - Provisioning transactions for system resources (for example, a request to create a Folder) use a synchronous request and a synchronous response.
  - Provisioning transactions for remote resources (for example, a request to create an Agent) use a synchronous request and an asynchronous response for maximum scaling. This means that a remote resource will go through a Pending state (accepted, but queued) before it is successfully provisioned and enters the Ready state (exists on underlying equipment and can be used).
- A change to a remote resource or one of its associated members makes that resource unavailable for any other changes until that change has propagated through the underlying equipment. This avoids race conditions and other system complexities.
- Provisioning requests are built around a number of simple verbs; create, update, delete, save, move, describe, retrieve, search and audit.
- If the client is using the SOAP protocol, the client can subscribe to be notified whenever changes to resources are detected. This is particularly important when local native interfaces can be used to change the underlying equipment, thus rendering any overlaying data model as out-of-date.

4.1.3 Item Types

The Unified CCDM Resource Management Web Service API has a class hierarchy that represents the resources and memberships available on the system. The following item types are supported:

- provisionable remote resource types (see section 3.1.2 "Provisionable Remote Resource Types")
- non-provisionable remote resource types (see section 3.1.3 "Non Provisionable Remote Resource Types")
- provisionable system resource types (see section 3.2.2 "Editable System Resource Types")
- non-provisionable system resource types (see section 3.2.3 "Read-Only System Resource Types")
- membership types (see section 4.7 "Resource Management Web Service Member Types").

4.1.4 Pkey Map Classes

The Unified CCDM Resource Management Web Service API also provides a *pkey map* class that links a resource with its corresponding resource on the remote equipment. For example, an Agent pkey map links the agent details in Unified CCDM with the agent details on the remote equipment.

The following resource types can have pkey maps:

- provisionable resources (for example, Agents, Agent Teams, Dialed Numbers)
- non-provisionable resources (for example, Announcements, Application Gateways, Bucket Intervals)
- member types (for example, Agent Agent Team Members, Agent Skill Group Members, Dialed Number Call Type Members)

Pkey maps can be used to link remote resources to multiple remote equipment instances with different details on each. For example, a single Agent resource may be located on both a Unified CCE and a Unified Communications Manager, with different login details on each. The agent pkey map links the Unified CCDM agent with the remote agent on each equipment instance and stores the different login details for each.

Pkey maps are also used to link resource memberships to the corresponding relationships on remote equipment. For example, if an Agent is in an Agent Team then the Agent Agent Team Member pkey map links the Unified CCDM membership with the corresponding agent to agent team relationship on the remote equipment.

Note

If you do not need to map resources to multiple equipment instances then you can ignore the pkey map resource and member classes although they are still present internally. All the information about a single equipment mapping is available directly from the resource or member class.

The exception is the `Describe()` method, since, if a field may be specified in a pkey map, the metadata about that field is returned against the pkey map type, not the corresponding resource type.

System resources (for example, Folders, Users, Groups) do not have pkey maps, as they do not exist on the remote equipment.

4.1.5 Methods

The Unified CCDM Resource Management Web Service API exposes the following methods:

- `Create()`
- `Update()`
- `Delete()`
- `Save()`
- `Move()`
- `Retrieve()`
- `Search()`
- `Describe()`
- `Audit()`

Depending on the requirements of the client these methods may be used together to provide required functionality. For example, to select from a list of Agent Team resources and then display the list of Agents within the selected Agent Team, you may:

1. Retrieve - Items of type Agent Team
2. <await user selection>
3. Retrieve - Agent Members of the selected Agent Team.

4.1.5.1 About Enterprise-Level Caching

Due to the distributed environment in which the Unified CCDM is deployed, some APIs rely on an enterprise level caching mechanism to provide enhanced scalability and response times. This behavior may mean that when polling services in quick succession data may take some time to update across all servers in the cluster and between the result sets of each API.

Table 4.1 shows which APIs provide direct database level access and which APIs rely on cached data from the enterprise level cache.

Table 4.1 Direct and Cached APIs

API	Direct Database Access	Enterprise Level Cache
`Create()`	Yes	
`Update()`	Yes	
`Delete()`	Yes	
`Save()`	Yes	
`Move()`	Yes	

API	Direct Database Access	Enterprise Level Cache
`Retrieve()`		Yes
`Search()`		Yes
`Describe()`		Yes
`Audit()`		Yes

Notifications sent from the Unified CCDM application server also rely on updates from the enterprise level cache so will only be sent when this cache is updated.

4.2 Resource Management Web Service Specification

4.2.1 SOAP

The WSDL file for the Resource Management Web Service is located on the Web/Application server here:

https://<server>:8085/ResourceManagement?wsdl

where **<server>** is the name of the Web/Application server.

4.2.2 REST

The REST protocol for each type and API is given in the section describing the type or API.

4.2.3 Errors

Table 4.2 Unified CCDM Resource Management Web Service API Errors

Error Code	Identifier	Description
1	`Exception`	General exception; represents an Operating System fault, for example, a Microsoft WCF web service internal error.
2	`ApplicationException`	General catch all application error; typically represents an unspecified fault in third party libraries if not explicitly trapped.
40	`ArgumentException`	Error(s) detected in the parameters passed in the web service request.

Error Code	Identifier	Description
50	SerializationException	Error(s) detected in the data structure passed in the web service request. This can either be missing or erroneous fields or a data type conversion in the contents of a field(s).
100	ConcurrencyConflict Exception	The resource to be updated has already been updated by another process, that is, its ChangeStamp field is too old.
101	ServiceNotReady Exception	The system service(s) are not yet ready to accept web service requests.
500	InvalidKeywordException	Search request error: the system does not recognize the supplied search keyword term, for example, specifying tipe rather than type.
501	InvalidCondition Exception	Search request error: the system could not parse the supplied search term condition, for example, specifying type:Aggent rather than type:Agent.
502	FieldConversion Exception	System could not parse the contents of a field. This is an application level error related to the system SerializationException and means that although the data type is correct, its value is incorrect in that context. For example, in: Agent.PeripheralName : 12~@FG 12~@FG is a valid string but an invalid value for this field type.
Various	EntityValidation Exception	The system has detected an error in the information supplied in the web service call. This is not an OS or field type error but an error in the composite data making up the whole request.
Various	ExonyApplication Exception	Last try application exception.
50150	NoMembership EffectivenessOverlap	A type of EntityValidationException error. Indicates that there is an error in the supplied member effective date-time range where the effective_from is newer than the effective_to.
50159	RequiredFieldNot Specified	A type of EntityValidationException error. Indicates that the contents of a field have been incorrectly sent as empty. For example, an Person.LoginName field has been supplied as an empty string.
50160	RegularExpression ValidationFailed	A type of EntityValidationException error. Indicates that the contents of a field have failed a check against the regular expression mask used to validate its contents. For example, an Agent.PeripheralName field does not match the mask needed for MAC addresses.

Error Code	Identifier	Description
50161	FieldValueOutOfRange	A type of EntityValidationException error. Indicates that the contents of a field are outside the boundary limits.
100000	MissingMember	Future use. Indicates that a mandatory member was not supplied in the web service.
100001	AgentMultiplePersonality	Future use.
100002	PeripheralRequired	Future use.
100003	RoutingClientRequired	Future use.
100005	UnknownClusterResource	A type of ExonyApplicationException error. Indicates that the supplied cluster resource type is not a supported type.
100006	AgentsInTeamCapacity Exceeded	Future use.
100007	SkillgroupIPTA	A type of EntityValidationException error. Indicates that for voice, skill groups must let the system pick the agent.
100039	CannotFindDefault RoutingClient	A type of EntityValidationException error. Indicates that the tenant referred to by the supplied folder id does not have access to the correct type of routing client needed for this type of request.
100040	CannotFindDefault Peripheral	A type of EntityValidationException error. Indicates that the tenant referred to by the supplied folder id does not have access to the correct type of peripheral needed for this type of request.
100041	CannotFindDefault OutboundRoutingClient	A type of EntityValidationException error. Indicates that the tenant referred to by the supplied folder id does not have access to the correct type of outbound routing client needed for this type of request.
100050	CantUpdateMaxAttempts	A type of EntityValidationException error. Indicates that the system has rejected an attempt to update a campaign when it is running.
100054	TooLongSkillGroupName	A type of EntityValidationException error. A special case of InternalName validation failure, where the associated Peripheral has a sub-skill group configured in which case the usual limit is reduced to 28 characters.
100058	LoginNameChangeNot Supported	A type of EntityValidationException error. Indicates that the supplied equipment login name cannot be provisioned at this time, typically due to configuration on the equipment.

Error Code	Identifier	Description
100074	NoEquipmentMappedTo Tenant	A type of EntityValidationException error. Indicates that the system has not found a cluster resource or resources of the correct type to match the supplied web request. For example, a create call type request has been supplied but a Unified CCE matching that request has not been found. This can happen if the cluster type instance has not been added by the ICE tool (See *Administration Guide for Cisco Unified Contact Center Domain Manager*) or the user does not have the correct security permissions to get access to it.
100097	BadRequestException	The requested action will cause a Precision Queue to have no Precision Queue Steps. A Precision Queue must have at least one Precision Queue Step.
250000	EnterpriseNameAlready ExistsException	The system has detected that a resource.InternalName for this resource type and equipment cluster already exists. This is an alternative primary key that must be unique.
250001	LoginNameAlreadyExists Exception	The system has detected that a User.LoginName already exists. The LoginName must be globally unique across the installation.

4.2.4 Restrictions

4.2.4.1 Array Limits

The Unified CCDM Resource Management Web Service APIs impose limits on the size of the array parameters that can be passed to protect the server and the clients. These limits are:

Table 4.3 Resource Limits

Method	Input Limit	Output Limit
Audit()	255 Resource Keys	255 Resource Audits
Create()	100 Resources or Resource Members	255 Request Results
Update()	100 Resources or Resource Members	255 Request Results
Delete()	255 Resource Keys	255 Request Results
Save()	100 Resources or Resource Members	255 Request Results
Move()	100 Resource Keys	255 Request Results
Describe()	None	None

Method	Input Limit	Output Limit
`Retrieve()`	255 Resource Keys	255 Resources
`Search()`	None	255 Resources

4.2.4.2 REST URI Segment Limit

In a REST URI, no individual segment between adjacent forward slash (/) delimiters can contain more than 255 characters. Query parameters are included in the character count, so this limit is most likely to be a consideration for the final segment which may contain several query parameters.

If the segment character limit is exceeded, the Web Service stack will return an HTML error page containing an Endpoint not Found error.

4.2.5 Field Validation

The Resource Management Web Service validates all supplied fields for correctness in terms of legal characters and length. However, validating these fields on the client before making the API call will improve the end customer experience and avoid round tripping for errors. The following table shows the common fields and their validation criteria.

Table 4.4 Common Field Validation

Resource	Field	Regular Expression	Length
Various	Internal Name	`^[a-zA-Z0-9][a-zA-Z0-9_\-\.\s]*$`	32
Agent	Extension	`^[a-zA-Z0-9]+$`	32
Agent	Peripheral Name	`^[a-zA-Z0-9][a-zA-Z0-9_\.]*$`	32
Agent Desk Top	Device Port Address	`^[a-zA-Z0-9][a-zA-Z0-9_\.]*$`	32
Dialed Number	Dialed Number String	`.*`	32
Various	MAC Address	`^([0-9a-fA-F][0-9a-fA-F]){5}([0-9a-fA-F][0-9a-fA-F])$`	12
Person	First/Last Name	`^[a-zA-Z0-9][a-zA-Z0-9_\-\.\s]*$`	32

Resource	Field	Regular Expression	Length
Person	Peripheral Login Name	`^[a-zA-Z0-9][a-zA-Z0-9_\.]*$`	32
Expanded Call Variable / User Variable	Name/Internal Name	`^[a-zA-Z0-9][a-zA-Z0-9_\-\.\s]*$`	27

4.3 Resource Management Web Service Data Types

This section lists common data types used in the Resource Management Web Service APIs. The data types in this section provide type-safe constructs for standard entities.

4.3.1 The Resource Hierarchy

The item and member types are represented by the resource class hierarchy shown in Figure 4.1 " Resource Class Hierarchy".

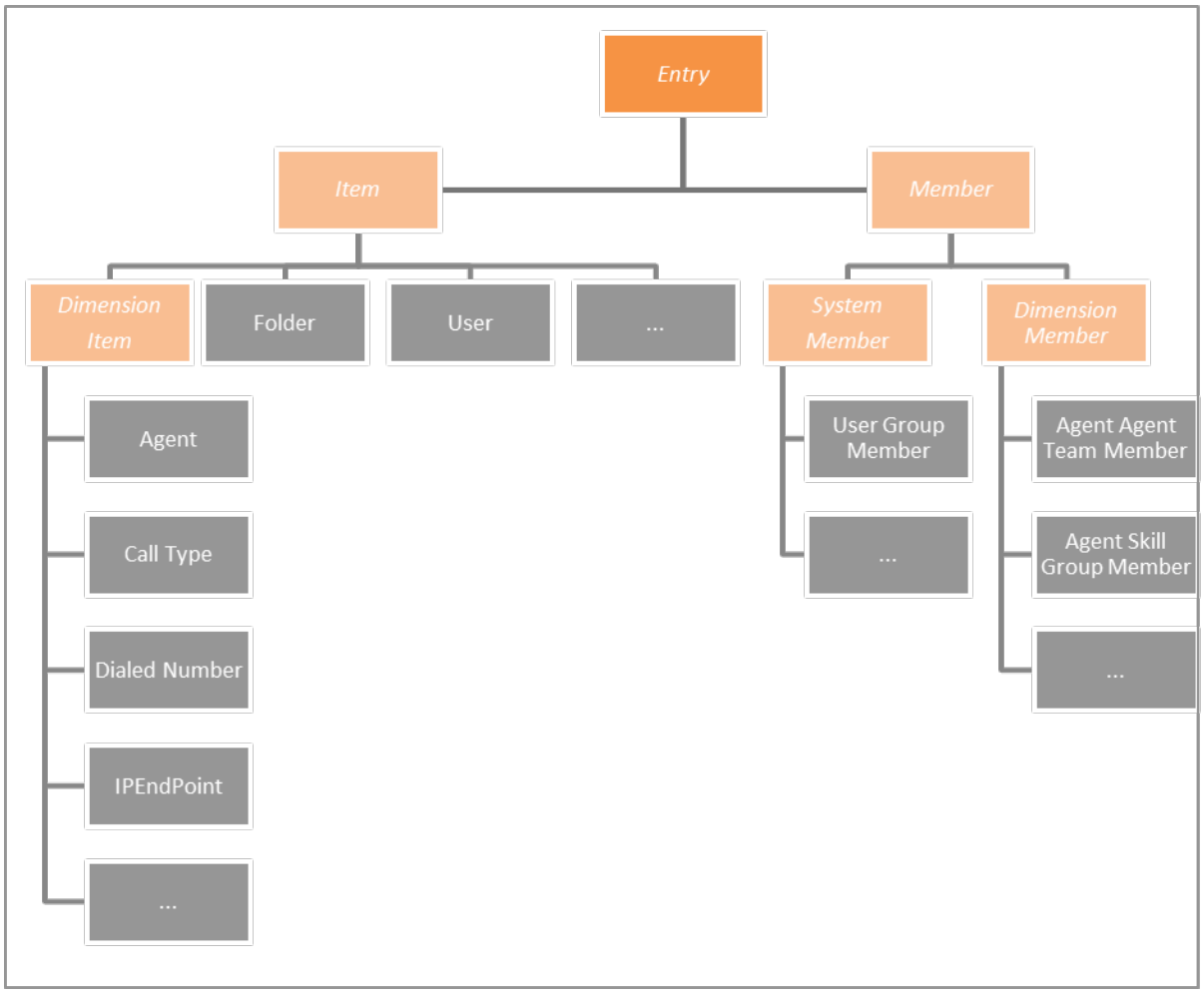


Figure 4.1 Resource Class Hierarchy

The top level abstract class, `Entry`, provides a simple-type definition for an entity. Requests to retrieve contact center objects such as Agents and Skill Groups take the form of these common objects.

An entity can be:

- an `Item`
- a `Member` which represents a membership between two items, and is one of:
  - a physical many-to-many relationship: for example, an Agent may be skilled in one or more Skill Groups
  - a type-2 SCD relationship, for example, an Agent may be part of an Agent Team during May 2010 and be part of a different Agent Team from June 2010 to Feb 2011.

Tip
Type-2 SCD relationships are rarely used in provisioning and should usually be filtered from search queries using the `latest:1` search term.

4.3.2 Common Fields and Classes

4.3.2.1 Name-Value Pair

The `NameValuePair` class is used to provide type-safe collections of information to be communicated to and from the server.

In the class descriptions in the rest of this chapter, items that are encoded as name-value pairs rather than named elements are denoted by (n-v) after the data type.

Properties

The exposed properties on the `NameValuePair` class are:

Table 4.5 Name-Value Pair Fields

Element Name	Data Type	Description	Required?
`Name`	`String`	The name field for the name-value pair.	Yes
`Value`	`String`	The value field for the name-value pair. To remove a name-value pair that was defined with a Custom storage type, set this field to the string **##REMOVE##**.	Yes

Element Name	Data Type	Description	Required?
StorageType	NameValuePair StorageType	The storage type for this name-value pair. One of: • **Standard**. Used by Unified CCDM • **Custom**. Used by the client application to store additional custom properties, for example, a custom property may be used to store an employee number for an Agent resource.	No (defaults to storage type Standard).

Note

The order of the elements in the NameValuePair class is significant. The elements must be supplied to the Web Service APIs in the order above, for both SOAP and REST protocols.

4.3.2.2 Status

Remote Resources

The status field is used to report the status of a remote resource. Figure 4.2 "Remote Resource State Transitions" shows the state transitions for a remote resource.

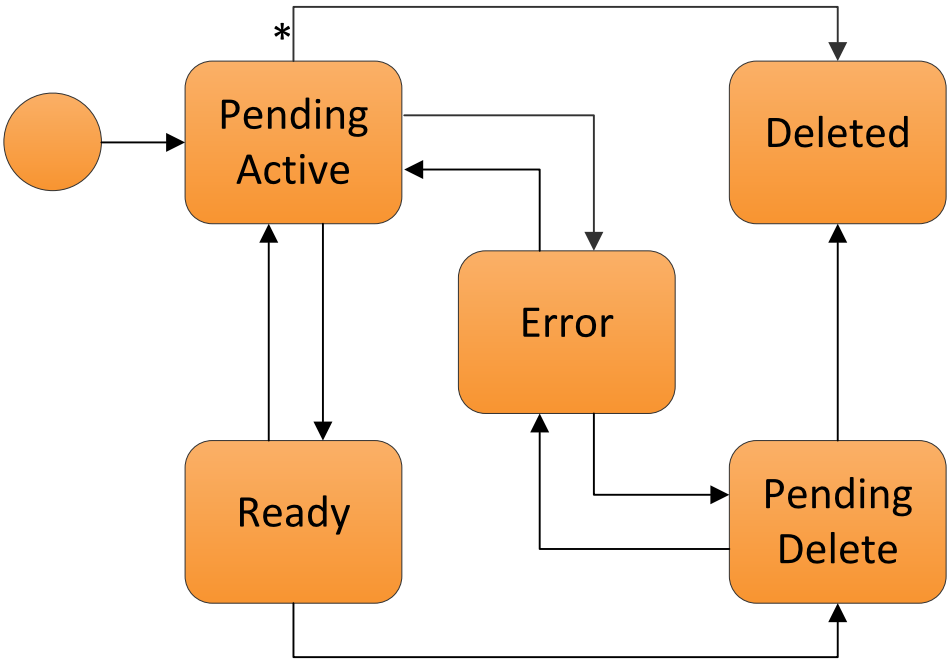


Figure 4.2 Remote Resource State Transitions

The valid status values are:

Status	Code	Description
Ready	R	The item is ready for use, both locally and on the remote system (for example, on Unified CCE).
Pending Active	S	(Or Synchronizing). A change has been made locally to this item that has not yet been reflected on the remote system. Newly created items start this state, before changing to Ready when they have been created on the remote system (for example, on Unified CCE).
Pending Delete	P	The item has been deleted from Unified CCDM but has not yet been removed from the remote system. Once the delete operation is complete on the remote system, the status changes to Deleted.

Status	Code	Description
Deleted	D	The item has been deleted on the remote system.
Error	E	An error has occurred while changing the item on the remote system. The current state in Unified CCDM does not reflect the state on the remote system. Errors should be corrected and the item re-submitted if required.

System Resources

The `status` field also applies to system resources, but system resources are not provisioned to a remote system, so do not have Pending Active, Pending Deleted or Error states. Figure 4.3 "System Resource State Transitions" shows the state transitions for system resources.

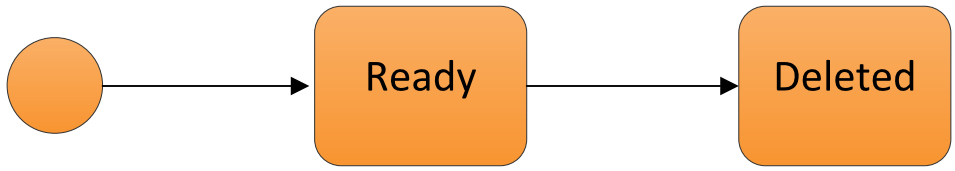


Figure 4.3 System Resource State Transitions

Note
ISE-enabled users are different. They follow the state transitions shown in Figure 4.2 "Remote Resource State Transitions" because changes to the linked Unified CCE user need to be provisioned. These additional state transitions apply when the ISE-enabled option is set or cleared, and when certain other changes are made to an ISE-enabled user.

4.3.3 Resource Hierarchy Classes

This section describes the classes that make up the resource hierarchy.

4.3.3.1 Entry

The `Entry` class represents a primary Unified CCDM entity or membership.

Properties

The exposed properties on the `Entry` class are:

Table 4.6 Entry Fields

Element Name	Data Type	Description	Required?
`Identity`	`String`	The identity of the related entity. An integer value for dimension entities or a GUID value for system entities. When creating a new resource, set this to -1.	Yes
`Type`	`String`	A string identifying the item type. One of the following: • a remote resource type from section 3.1 "Remote Resource Types" • a system resource type from section 3.2 "System Resource Types" • a member type from section 4.7.1.2 "Member Types and Identifiers" • a pkey type from section 4.3.5.3 "Provisionable Remote Resource Pkey Map Types" or section 4.3.5.4 "Non Provisionable Remote Resource Pkey Map Types" • a pkey member type from section 4.3.5 "Pkey Map Classes".	Yes
`EffectiveFrom`	`DateTime`	The UTC date that the entity will become effective or has become effective in the contact center environment.	Yes
`EffectiveTo`	`DateTime`	The UTC date that the entity will expire or has expired and been removed or deactivated in the contact center environment.	Yes
`Status`	`String`	The provisioning status of the resource. See section 4.3.2.2 "Status".	Yes
`Changestamp`	`Int`	The concurrency change stamp for this resource. For provisionable resources, the system checks the value of this field to determine whether the provisioning request is allowed, that is, whether another client has updated the resource since it was retrieved or searched for. Not required for `Create()`, but must be supplied for `Update()`.	Yes (for `Update()`)
`Fields`	`NameValuePair[]`	A collection of the fields that have been set or may be set on the current resource.	Yes

Element Name	Data Type	Description	Required?
EquipmentMapping	Resource[]	An array containing either ItemPkey or MemberPkey elements, specifying resource details that are specific to an equipment instance. Only required if one of the following is true: • Type is one of the resource types listed in section 3.1 "Remote Resource Types" and the resource is mapped to more than one equipment instance. • Type is one of the resource membership types listed in section 4.7.1.2 "Member Types and Identifiers" and the membership is mapped to more than one equipment instance. For more information about the use of this field when creating resources and memberships, see section 4.8.1.6 "Creating Items with Pkey Maps". For more information about the use of this field when retrieving resources and memberships, see section 4.8.6.6 "Retrieving Items with Pkey Maps".	No
Custom	Int	User-defined custom data.	No
CreatedById	Guid (n-v pair)	Read-only id of the user that created this item, returned on retrieve or search responses.	n/a
ModifiedById	Guid (n-v pair)	Read-only id of the user that last modified this item, returned on retrieve or search responses.	n/a
CreationDate	DateTime (n-v pair)	Read-only creation date for this item, returned on retrieve or search responses.	n/a
ModifiedDate	DateTime (n-v pair)	Read-only last modified date for this item, returned on retrieve or search responses.	n/a
Deleted	Bool (n-v pair)	Read-only flag indicating whether the item has been deleted.	n/a
Hidden	Bool (n-v pair)	Read-only flag indicating whether the item has been hidden.	n/a
Latest	Bool (n-v pair)	Read-only flag indicating whether this resource is the latest in a type-2 sequence.	n/a
System	Bool (n-v pair)	Read-only flag indicating whether the item is owned by the system.	n/a

Note
The order is significant for the elements in the `Entry` class that are not supplied as name-value pairs. These elements must be supplied to the Web Service APIs in the order above, for both SOAP and REST protocols.

4.3.3.2 Item

The `Item` class represents an entity that can be created in Unified CCDM or imported into Unified CCDM. An Item is always has a name, and a location in the security folder tree.

Properties

The exposed properties on the `Item` class are:

Table 4.7 Item Fields

Element Name	Data Type	Description	Required?
Inherits from `Entry`			
`FolderId`	`Guid` (n-v pair)	The identity of the related folder.	Yes
`Name`	`String` (n-v pair)	This is the friendly name for this resource, typically used for reporting processes.	Yes
`InternalName`	`String` (n-v pair)	The name of the entity on underlying equipment. For example, in Unified CCE this represents the Cisco `EnterpriseName` field. If the resource exists on multiple equipment instances, this is the name of the entity on the primary cluster resource (defined as the first cluster on which the item is provisioned).	No
`Description`	`String` (n-v pair)	The description of this instance of the resource.	No

Element Name	Data Type	Description	Required?
MappedClusterResources	String (n-v pair)	The identifier of the underlying equipment on which the resource exists. If the resource exists on multiple equipment instances, this is a comma-separated list of the equipment instances. Default: System will select based on Tenant.	No
MappedClusterResourceDetails	String (n-v pair)	The details of the underlying equipment on which the resource exists. If the resource exists on multiple equipment instances, this is a comma-separated list of the details for each equipment instance.	No

4.3.3.3 Dimension Item

The DimensionItem class represents a reporting dimension or a provisionable entity. It may be created in Unified CCDM or imported into Unified CCDM.

The exposed properties on the DimensionItem class are:

Table 4.8 Dimension Item Fields

Element Name	Data Type	Description	Required?
Inherits From: Item			
HierarchyParentUrn	Int (n-v pair)	Read-only. The type-1 Hierarchy for this dimension item. Always -1.	No
ItemBusinessUrn	Int (n-v pair)	Read-only. Type-2 Business id for this dimension item. Default: -1.	No
OwnerId	Guid (n-v pair)	Read-only. The GUID of the tenant dimension folder that contains this dimension.	No
TenantBizUrn	Int (n-v pair)	Read-only. Item Business Urn of the tenant dimension associated with this dimension.	No
TimeZoneUrn	Int (n-v pair)	Read only. The time zone in which the resource is located.	No

4.3.3.4 Member

The `Member` class represents a membership between two entities. One of the entities in the membership will be designated the parent entity, and the other the child. Often, the relationship will be many-many, but typically, the parent member will have many more children than the child member has parents. For example, in a Skill Group to Agent relationship, the Skill Group is the parent and the Agent the child. A Skill Group can contain a number of Agents, and an Agent can belong to a number of Skill Groups, but typically, a Skill Group will contain many Agents, whereas an Agent will belong to a just a few Skill Groups,

Properties

The exposed properties of the `Member` class are:

Table 4.9 Member Fields

Element Name	Data Type	Description	Required?
Inherits From: `Entry`			
`ParentId`	`Int` (n-v pair)	This is the id of the resource which is the parent resource for the membership.	Yes
`ChildId`	`Int` (n-v pair)	This is the id of the resource which is the child resource for the membership.	Yes

4.3.3.5 Dimension Member

The `DimensionMember` class inherits from the `Member` class and represents a membership between `Dimensionitem` (or remote resource) entities. A key characteristic is that these memberships normally have type-2 effective dating enabled.

Properties

The exposed properties on the `DimensionMember` class are:

Table 4.10 Dimension Member Fields

Element Name	Data Type	Description	Required?
Inherits From: `Member`			
`LatestParentItemUrn`	`Int` (n-v pair)	This is the id of the resource which is the most recent parent resource for the membership.	Yes
`LatestChildItemUrn`	`Int` (n-v pair)	This is the id of the resource which is the most recent child resource for the membership.	Yes

4.3.3.6 System Member

The `SystemMember` class inherits from the `Member` class and represents the membership between system resource entities.

The exposed properties on the `SystemMember` class are:

Table 4.11 System Member Fields

Property	Data Type	Description	Required
Inherits From: `Member`			
`LatestParentItemUrn`	`Int` (n-v pair)	This is the id of the resource which is the most recent parent resource for the membership.	Yes
`LatestChildItemUrn`	`Int` (n-v pair)	This is the id of the resource which is the most recent child resource for the membership.	Yes

4.3.4 Other Common Classes

This section describes other common classes used by the Resource Management Web Service APIs.

4.3.4.1 Resource Meta

The `ResourceMeta` class describes a field for an entity of a given type.

Properties

The exposed properties on the `ResourceMeta` class are:

Table 4.12 Resource Meta Properties

Property	Data Type	Description
Name	String	The name of the field that may be set/read on the entity
DataType	String	The data type of the field that may be set/read on the entity
Length	Int	The length of the data type.
Required	Bool	States if the field described is required or optional
DefaultValue	String	The default value as a string for the field that has been specified
CanCreate	Bool	Determines whether this field is required or valid for a create operation.
CanEdit	Bool	Determines if the field can be changed as part of an update operation.
FieldGrouping	String	Indicates how this field can be grouped for client UIs or workflows. One of: **Common**. The field is common to all resource types **Specific**. The field is specific to this resource type.
FieldType	String	Indicates the usage of this field for automated workflow clients. One of: **Key**. Field contains an Id or Urn to another resource instance. **Field**. The field is an attribute of this type of resource. **Virtual**. The field is defined by a calculation or algorithm. **Xml**. The field contains an XML data block specific to this resource type. **Equipment**. The field contains equipment cluster reference.
HelpText	String	A description of the use of this field.

4.3.4.2 Resource Field Meta

The `ResourceFieldMeta` class describes the associated fields for an entity of a given type.

Properties

The exposed properties on the `ResourceFieldMeta` class are:

Table 4.13 Resource Field Meta Properties.

Property	Data Type	Description
Type	String	The resource type to which this metadata description applies. One of the following resource types: • a remote resource type from section 3.1 "Remote Resource Types" • a system resource type from section 3.2 "System Resource Types" • a member type from section 4.7.1.2 "Member Types and Identifiers" • a pkey type from section 4.3.5.3 "Provisionable Remote Resource Pkey Map Types" or section 4.3.5.4 "Non Provisionable Remote Resource Pkey Map Types" • a pkey member type from section 4.3.5 "Pkey Map Classes".
CanCreate	Bool	An instance of this resource may be created in the Unified CCDM database.
CanEdit	Bool	An instance of this resource may be edited.
CanDelete	Bool	An instance of this resource may be deleted from the Unified CCDM database.
CanProvision	Bool	An instance of this resource will be provisioned into one or more equipment native objects on the underlying CC equipment.
IsHidden	Bool	An instance of this class may be hidden, that is, reserved for internal use by Unified CCDM only.
IsSystem	Bool	An instance of this class may be manipulated only by user accounts of Host Administrator only.
ClusterResourceTypes	String	A comma separated string of the underlying equipment types supported by this resource.
Fields	ResourceMeta[]	The description of all the fields that make up this entity type.

4.3.4.3 Resource Key

The ResourceKey class represents the entity identity when performing a delete, retrieve, search or audit operation.

Properties

The exposed properties on the ResourceKey class are:

Table 4.14 Resource Key Properties

Property	Data Type	Description
Identity	String	The identity of the related item. May be an integer value for dimension entities or a GUID value for system entities.
ResourceType	String	The item type.

4.3.4.4 Resource Audit

The ResourceAudit class describes an audit record that has been logged as part of a provisioning request.

Properties

The exposed properties on the ResourceAudit class are:

Table 4.15 Resource Audit Properties

Property	Data Type	Description
AuditType	String	The type of audit (one of "System" or "Resource").
EventDateTime	String	The UTC Date Time at which the audited event was entered into the system.
EventOutcome	String	The outcome of the audited operation. One of: S: Success, F: Failure, N: Not applicable, U: Unknown.
UserName	String	The name of the user that caused the creation of the audit. May be System if the audit operation was performed by a Unified CCDM scheduled process such as Data Importer.
Description	String	The text indicating which provisioning operation took place.
ResourceIdentity	String	The identifier of the resource that is the subject of the audit. Together with the ResourceType is the unique composite identifier of the resource.
ResourceType	String	The type of the resource that is the subject of the audit. Together with the ResourceIdentity, this is the unique composite identifier of the resource.
ResourceName	String	The name of the resource that is the subject of the audit.
Detail fields:		
UserAgentName	String	The name of the user agent that invoked the request that caused the audited provisioning request.

Property	Data Type	Description
ClusterResource	String	The remote cluster(s) on which the provisioning operation occurred.
ChangedFields	XmlElement	The fields that were modified during the provisioning request.
AdditionalData	XmlElement	Any additional supporting data for this audited event.

4.3.4.5 Request Result

The RequestResult class holds the status and identity information for a given change. This object contains the identity of any item that was created, deleted or updated, the new status of the item and a collection of error messages if exceptions occurred when the item was changed.

Properties

The exposed properties on the RequestResult class are:

Table 4.16 Request Result Properties

Property	Data Type	Description
Identity	String	The identity of the item that was created or changed.
Type	String	The item type.
Name	String	The name of the resource. Only valid if Type is not a member type.
Status	String	The status of the resource that has been changed. See section 4.3.2.2 "Status"
Errors	String []	The error messages returned as a result of changing this item. If no errors occurred, this is an empty array.

4.3.4.6 Resource Fault Detail

The ResourceFaultDetail class is returned when any method fails and represents the details logged by an exception or other error detected by the server.

Properties

The exposed properties on the ResourceFaultDetail class are:

Table 4.17 Resource Fault Detail Properties

Property	Data Type	Description
`Message`	`String`	A message describing the error that occurred.
`ErrorType`	`String`	The error type. One of: • **Validation**: the request is invalid. • **Format**: the request format is invalid. • **System**: the system state is invalid. • **General**: another error has occurred.
`ErrorCode`	`Int`	One of the error codes listed in section 4.2.3 "Errors".

4.3.4.7 Additional Data

The `AdditionalData` class is used by some APIs to send additional name-value fields.

Properties

The exposed properties on the `AdditionalData` class are:

Table 4.18 Additional Data Properties

Property	Data Type	Description
`Fields`	`NameValuePair[]`	An array of name-value fields.

4.3.4.8 Page Info

The `PageInfo` class is used by the `Audit()` API to hold paging information about the audit results being returned.

Properties

The exposed properties on the `PageInfo` class are:

Table 4.19 Page Info Properties

Property	Data Type	Description
`ResultsPerPage`	`Int`	The number of results to be returned in each page of results.
`StartIndex`	`Int`	The index of the first element to be returned.
`TotalResults`	`Int`	The total number of audit results available to be returned.

4.3.4.9 Resource Audit Results

The `ResourceAuditResults` class is used by the `Audit()` API to return audit results.

Properties

The exposed properties on the `ResourceAuditResults` class are:

Table 4.20 Resource Audit Results Properties

Property	Data Type	Description
`PageInfo`	`PageInfo`	Information about the collection of audit results being returned (see section 4.3.4.8 "Page Info").
`Audits`	`ResourceAudit[]`	A collection of `ResourceAudit` objects containing audit data for the resource id passed in (see section 4.3.4.4 "Resource Audit ").

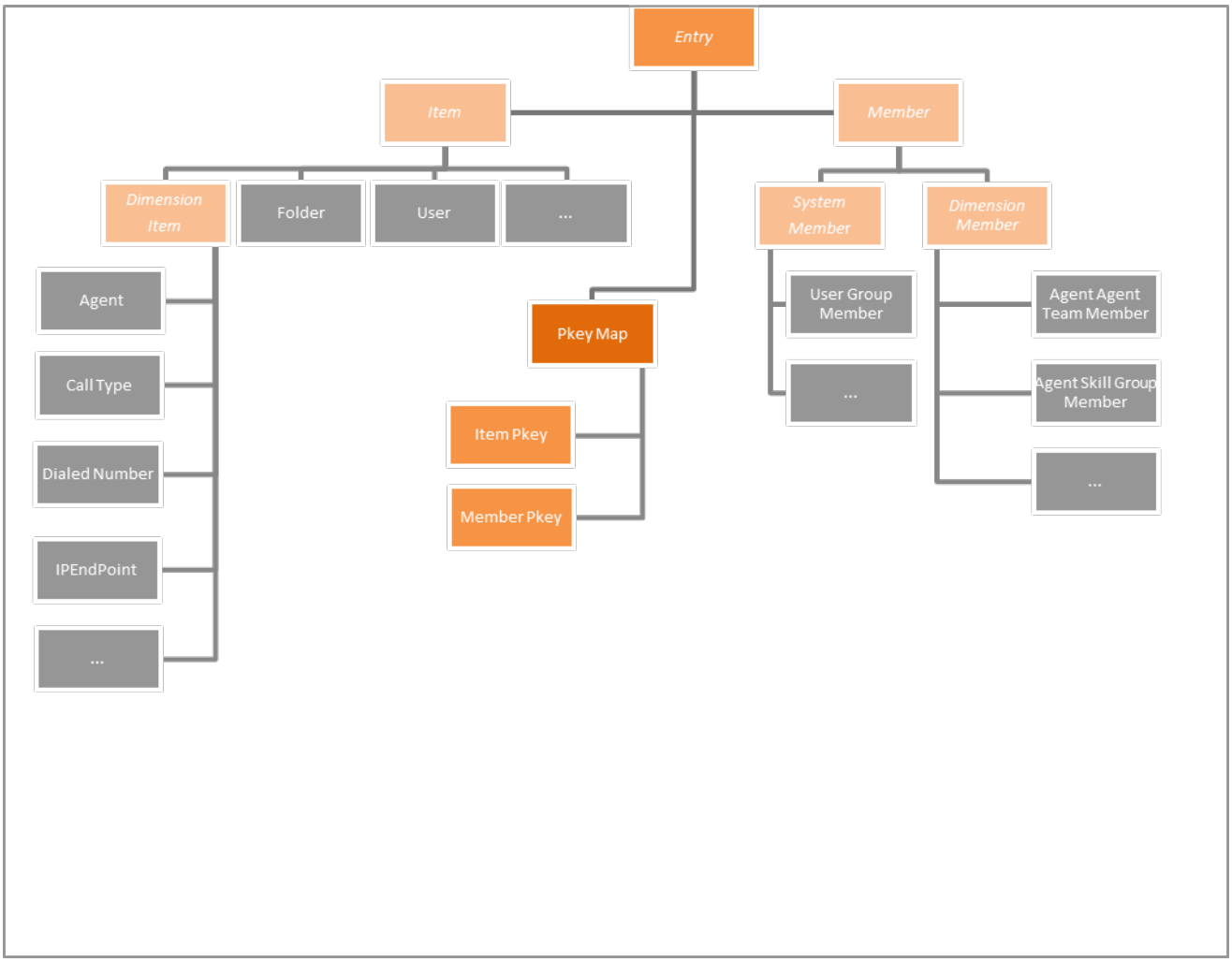
4.3.5 Pkey Map Classes

4.3.5.1 About Pkey Map Classes

A pkey map links a resource or membership with its corresponding resource or membership on the remote equipment. Pkey maps can be used to link remote resources and memberships to multiple remote equipment instances with different details on each.

You do not need to use the pkey map classes unless you need to map resources to multiple equipment instances. All the information about a single equipment mapping is available directly from the resource or member class.

4.3.5.2 Resource Hierarchy Including Pkey Maps



4.3.5.3 Provisionable Remote Resource Pkey Map Types

These remote resource pkey map types are fully supported by the Resource Management Web Services.

Resource Pkey Map Type	Internal Name	REST Parameter
Agent Pkey	PK_AGENT_PKEY	agent-pkey
Agent Desktop Pkey	PK_AGENT_DESKTOP_PKEY	agent-desktop-pkey

Resource Pkey Map Type	Internal Name	REST Parameter
Agent Team Pkey	PK_AGENT_TEAM_PKEY	agent-team-pkey
Call Type Pkey	PK_CALL_TYPE_PKEY	call-type-pkey
Calling Search Space Pkey	PK_CALLING_SEARCH_SPACE_PKEY	calling-search-space-pkey
Department Pkey	PK_DEPARTMENT_PKEY	department-pkey
Dialed Number Pkey	PK_DIALED_NUMBER_PKEY	dialed-number-pkey
Enterprise Skill Group Pkey	PK_ENTERPRISE_SKILLGROUP_PKEY	enterprise-skill-group-pkey
Expanded Call Variable Pkey	PK_EXPANDED_CALL_VARIABLE_PKEY	expanded-call-variable-pkey
Label Pkey	PK_LABEL_PKEY	label-pkey
Network VRU Script Pkey	PK_NETWORK_VRU_SCRIPT_PKEY	network-vru-script-pkey
Person Pkey	PK_PERSON_PKEY	person-pkey
Precision Attribute Pkey	PK_PRECISION_ATTRIBUTE_PKEY	precision-attribute-pkey
Precision Queue Pkey	PK_PRECISION_QUEUE_PKEY	precision-queue-pkey
Precision Queue Step Pkey	PK_PRECISION_QUEUE_STEP_PKEY	precision-queue-step
Route Pkey	PK_ROUTE_PKEY	route-pkey
Route Partition Pkey	PK_ROUTE_PARTITION_PKEY	route-partition-pkey
Service Pkey	PK_SERVICE_PKEY	service-pkey
Skill Group Pkey	PK_SKILLGROUP_PKEY	skill-group-pkey
Tenant Pkey	PK_TENANT_PKEY	tenant-pkey
User Variable Pkey	PK_USER_VARIABLE_PKEY	user-variable-pkey

4.3.5.4 Non Provisionable Remote Resource Pkey Map Types

These remote resource pkey map types are supported by the Resource Management Web Services for searching only.

Resource Pkey Map Type	Internal Name
Announcement Pkey	PK_ANNOUNCEMENT_PKEY
Application Gateway Pkey	PK_APPLICATION_GATEWAY_PKEY
Application Instance Pkey	PK_APPLICATION_INSTANCE_PKEY
Bucket Interval Pkey	PK_BUCKET_INTERVAL_PKEY

Resource Pkey Map Type	Internal Name
Call Source Pkey	PK_CALL_SOURCE_PKEY
Campaign Pkey	PK_CAMPAIGN_PKEY
Category Pkey	PK_CATEGORY_PKEY
Chargeband Pkey	PK_CHARGEBAND_PKEY
Cli Pkey	PK_CLI_PKEY
Custom Entry Pkey	PK_CUSTOM_ENTRY_PKEY
Date Time Setting Pkey	PK_DATE_TIME_SETTING_PKEY
Device Target Pkey	PK_DEVICE_TARGET_PKEY
Dialer Pkey	PK_DIALER_PKEY
Dial Number Plan Pkey	PK_DIAL_NUMBER_PLAN_PKEY
Enterprise Route Pkey	PK_ENTERPRISE_ROUTE_PKEY
Enterprise Service Pkey	PK_ENTERPRISE_SERVICE_PKEY
Gateway Function Pkey	PK_GATEWAY_FUNCTION_PKEY
Gateway Result Pkey	PK_GATEWAY_RESULT_PKEY
Gateway Server Pkey	PK_GATEWAY_SERVER_PKEY
ICR Instance Pkey	PK_ICR_INSTANCE_PKEY
Import Rule Pkey	PK_IMPORT_RULE_PKEY
IVR Entry Point Pkey	PK_IVR_ENTRY_POINT_PKEY
IVR Module Pkey	PK_IVR_MODULE_PKEY
IVR Routing Target Pkey	PK_IVR_ROUTING_TARGET_PKEY
IVR Script Pkey	PK_IVR_SCRIPT_PKEY
IVR Script Node Pkey	PK_IVR_SCRIPT_NODE_PKEY
Logical Interface Controller Pkey	PK_LOGICAL_INTERFACE_CONTROLLER_PKEY
Media Class Pkey	PK_MEDIA_CLASS_PKEY
Media Routing Domain Pkey	PK_MEDIA_ROUTING_DOMAIN_PKEY
Network Trunk Group Pkey	PK_NETWORK_TRUNK_GROUP_PKEY

Resource Pkey Map Type	Internal Name
Network Vru Pkey	PK_NETWORK_VRU_PKEY
Object Type Pkey	PK_OBJECT_TYPE_PKEY
Peripheral Pkey	PK_PERIPHERAL_PKEY
Physical Interface Controller Pkey	PK_PHYSICAL_INTERFACE_CONTROLLER_PKEY
Port Pkey	PK_PORT_PKEY
Query Rule Pkey	PK_QUERY_RULE_PKEY
Rating Period Pkey	PK_RATING_PERIOD_PKEY
Reason Code Pkey	PK_REASON_CODE_PKEY
Region Pkey	PK_REGION_PKEY
Route Partition Pkey	PK_ROUTE_PARTITION_PKEY
Routing Client Pkey	PK_ROUTING_CLIENT_PKEY
Routing Script Pkey	PK_ROUTING_SCRIPT_PKEY
Schedule Pkey	PK_SCHEDULE_PKEY
Scheduled Target Pkey	PK_SCHEDULED_TARGET_PKEY
Script Pkey	PK_SCRIPT_PKEY
Script Node Pkey	PK_SCRIPT_NODE_PKEY
Strategy Pkey	PK_STRATEGY_PKEY
Timeband Pkey	PK_TIMEBAND_PKEY
Tli Pkey	PK_TLI_PKEY
Trunk Pkey	PK_TRUNK_PKEY
Trunk Group Pkey	PK_TRUNK_GROUP_PKEY
Wrapup Code Pkey	PK_WRAPUP_CODE_PKEY

4.3.5.5 Member Pkey Map Types

These member pkey map types are fully supported by the Resource Management Web Services.

Member Pkey Map Type	Internal Name	REST Parameter
Agent Agent Team Member Pkey	PK_AGENT_AGENT_TEAM_MEMBER_PKEY	agent-agent-team-member-pkey
Agent Peripheral Member Pkey	PK_AGENT_PERIPHERAL_MEMBER_PKEY	agent-peripheral-member-pkey
Agent Precision Attribute Member Pkey	PK_AGENT_PRECISION_ATTRIBUTE_MEMBER_PKEY	agent-precision-attribute-member-pkey
Agent Skill Group Member Pkey	PK_AGENT_SKILLGROUP_MEMBER_PKEY	agent-skill-group-member-pkey
Call Type Routing Script Member Pkey	PK_CALL_TYPE_ROUTING_SCRIPT_MEMBER_PKEY	call-type-routing-script-member-pkey
Dialed Number Call Type Member Pkey	PK_DIALED_NUMBER_CALL_TYPE_MEMBER_PKEY	dialed-number-call-type-member-pkey
Precision Queue Step Precision Attribute Member Pkey	PK_PRECISION_QUEUE_STEP_PRECISION_ATTRIBUTE_MEMBER_PKEY	precision-queue-step-precision-queue-attribute-member-pkey
Query Rule Campaign Member Pkey	PK_QUERY_RULE_CAMPAIGN_MEMBER_PKEY	query-rule-campaign-member-pkey
Route Partition Calling Search Space Member Pkey	PK_ROUTE_PARTITION_CALLING_SEARCH_SPACE_MEMBER_PKEY	route-partition-calling-search-space-member-pkey
Skill Group Campaign Member Pkey	PK_SKILLGROUP_CAMPAIGN_MEMBER_PKEY	skill-group-campaign-member-pkey
Skill Group Service Member Pkey	PK_SKILLGROUP_SERVICE_MEMBER_PKEY	skill-group-service-member-pkey

4.3.5.6 Pkey Map Class

The PkeyMap class is used to map items to more than one equipment instance. If an item is associated with multiple equipment instances, there will be a pkey map associated with each equipment instance.

Properties

The exposed properties on the PkeyMap class are:

Table 4.21 Pkey Map Fields

Element Name	Data Type	Description	Required?
Inherits from Entry where Type is a pkey type (section 4.3.5.3 "Provisionable Remote Resource Pkey Map Types" or section 4.3.5.4 "Non Provisionable Remote Resource Pkey Map Types" or a member pkey type (section 4.3.5 "Pkey Map Classes")			
Purged	Bool (n-v pair)	Read-only flag indicating whether the Pkey item has been purged.	n/a
Pkey1	Guid (n-v pair)	Read-only flag containing the primary identifier of the item on the specified source.	n/a
Pkey2	Guid (n-v pair)	Read-only flag containing the secondary identifier of the item on the specified source.	n/a
ClusterResourceId	String (n-v pair)	The identifier of the equipment instance which this pkey map relates to.	No

4.3.5.7 Item Pkey Class

The ItemPkey class applies to resources that are mapped to more than one equipment instance. It represents resource information from the equipment instances that this resource is mapped to. It may be created in Unified CCDM or imported into Unified CCDM.

Properties

The exposed properties on the ItemPkey class are:

Table 4.22 Item Pkey Fields

Element Name	Data Type	Description	Required?
Inherits from `PkeyMap` where `Type` is a pkey type (section 4.3.5.3 "Provisionable Remote Resource Pkey Map Types" or section 4.3.5.4 "Non Provisionable Remote Resource Pkey Map Types")			
`InternalName`	`String` (n-v pair)	The name of the entity on the equipment instance which this pkey map relates to.	Yes
`Description`	`String` (n-v pair)	The description of the entity on the equipment instance which this pkey map relates to.	No

4.3.5.8 Member Pkey Class

The `MemberPkey` class applies to memberships that that are mapped to more than one equipment instance. It represents membership information from the equipment instances that this membership is mapped to. It may be created in Unified CCDM or imported into Unified CCDM.

Properties

The exposed properties of the `MemberPkey` class are:

Table 4.23 Member Pkey Fields

Element Name	Data Type	Description	Required?
Inherits From: `PkeyMap` where `Type` is a member pkey type (section 4.3.5 "Pkey Map Classes")			
`ParentId`	`Int` (n-v pair)	This is the id of the resource which is the parent resource for the member pkey.	Yes
`ChildId`	`Int` (n-v pair)	This is the id of the resource which is the child resource for the member pkey.	Yes

4.3.5.9 More Information

Specific information about using the pkey map classes is available as follows:

- Each of the provisionable resource types described in section 4.5 "Provisionable Remote Resource Types" includes information about the fields that may be located on pkey items.
- Each of the APIs described in section 4.8 "Resource Management Web Service APIs" explains how to use pkey maps with that API.

4.4 Editable System Resource Types

4.4.1 Folder Resource Item

4.4.1.1 Description

The `Folder` resource describes a location in the folder hierarchy that is a container for other system resources or remote resources.

4.4.1.2 Fields

For folders only the default Resource level fields can be specified. The `FolderId` field represents the identity of the parent folder. For example the path `/Tenant1/Folder1` is identified by a name of Folder1 and a `FolderId` of the Tenant1 folder.

Table 4.24 Folder Resource Item Fields

Element Name	Data Type	Description	Required?
Inherits From: `Item`			
`Path`	`String` (n-v pair)	The path defining the location of this folder. This is a read-only field set by Unified CCDM on Retrieve or Search operations only.	Yes
`ParentId`	`Guid` (n-v pair)	The identifier of the folder that is the immediate parent of this folder. This is a read-only field set by Unified CCDM on Retrieve or Search operations only.	Yes
`PolicyId`	`Guid` (n-v pair)	The id of the security policy associated with the folder. This is a read-only field set by Unified CCDM on Retrieve or Search operations only.	Yes
`FolderType`	`String` (n-v pair)	The type of folder represented by this instance. One of ‘F’ for folder or ‘T’ for tenant folder. Folders of type ‘T’ are created by the Create Tenant use case only.	Yes
`PolicyRoot`	`Bool` (n-v pair)	A flag indicating whether the security policy associated with the folder is the policy root. A value of 1 indicates root policy. This is a read-only field set by Unified CCDM on Retrieve or Search operations only.	Yes

Note

The `Folder` resource does not use the following `Item` elements: `InternalName`, `MappedClusterResources` and `MappedClusterResourceDetails`.

4.4.1.3 Associated Name Fields for Folders

As well as the fields above, the following associated name fields can also be used to sort Folder resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.25 Associated Name Fields for Sorting Folders

Associated Name Field	Related Element
`ParentCreatedByLoginName`	`Entry.CreatedById`
`ParentFolderPath`	`Item.FolderId`

4.4.1.4 Model

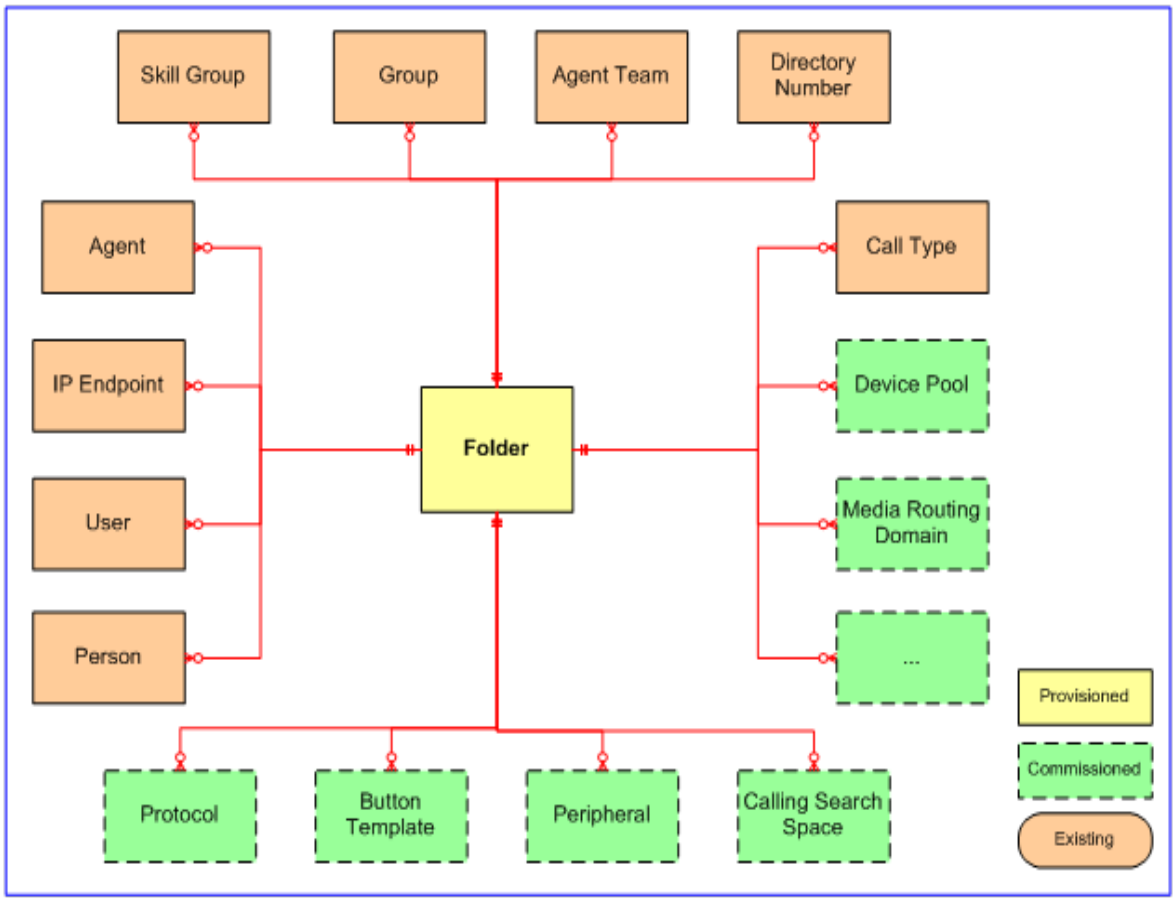


Figure 4.4 Folder Relationships

4.4.1.5 REST Protocol

Folder REST API Summary				
Base URL	https://<server>:8085/resourcemanagement/rest/resources			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL] /folders/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Sync.	POST	200 OK	[Base URL]/folders
Delete	Sync.	DELETE	202 Accepted	[Base URL]/folders/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/folder
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/folders/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/folders
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_FOLDER&max%3a10
Update	Sync.	PUT	202 Accepted	[Base URL]/folders/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/folders			

4.4.2 Group Resource Item

4.4.2.1 Description

The Group resource is used to group users to apply role based security. Groups can contain other groups.

4.4.2.2 Fields

Only the default Resource level fields can be set or read on a group resource.

Table 4.26 Group Resource Item Fields

Element Name	Data Type	Description	Required?
Inherits From: `Item`			
`Enabled`	`Bool` (n-v pair)	Flag indicating whether the group is enabled. Used to disable a security group.	No
`Hidden`	`Bool` (n-v pair)	Flag indicating whether the group is hidden. Used for system security groups that are not visible to tenants.	No

Note

The `Group` resource does not use the following `Item` elements: `InternalName`, `MappedClusterResources` and `MappedClusterResourceDetails`.

4.4.2.3 Associated Name Fields for Groups

As well as the fields above, the following associated name fields can also be used to sort Group resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.27 Associated Name Fields for Sorting Groups

Associated Name Field	Related Element
`ParentCreatedByLoginName`	`Entry.CreatedById`
`ParentFolderPath`	`Item.FolderId`

4.4.2.4 Model

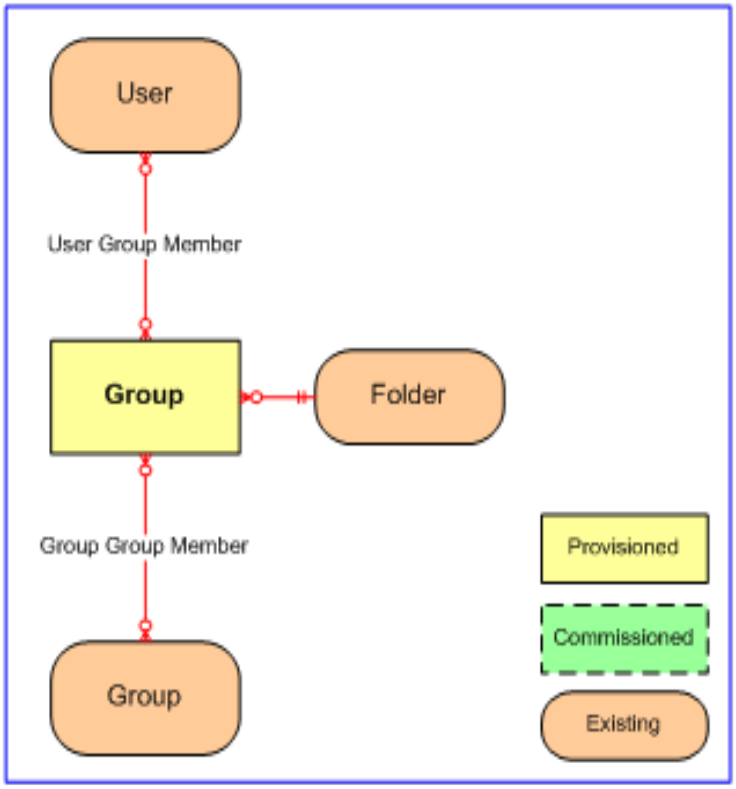


Figure 4.5 Group Relationships

4.4.2.5 REST Protocol

Group REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL] /groups/<id>,<id>/audits? fromDate=<fromdate>& toDate=<toDate>…`
Create	Sync.	`POST`	`200 OK`	`[Base URL]/groups`

Group REST API Summary				
Delete	Sync.	DELETE	202 Accepted	[Base URL]/groups/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/group
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/groups/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/groups
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_GROUP&max%3a10
Update	Sync.	PUT	202 Accepted	[Base URL]/groups/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/groups			

4.4.3 User Resource Item

4.4.3.1 Description

The `User` resource is the login record for a Unified CCDM user. It defines the user's association with roles, tasks and groups and determines what they are allowed to see and do. There are typically three classes of users:

- Host Administrator - can view and amend all resources on the complete platform.
- Tenant Administrator - can view and amend all resources that are owned by that tenant. This is the typical user account that calls the Unified CCDM Web Services.
- Agent or Supervisor - can view and amend all resources to which the tenant folder structure allows them, for example, Baltimore agents can only view resources in the Baltimore folder.

4.4.3.2 Fields

The following fields can be read or set for a user resource.

Table 4.28 Fields in User Resource Item

Element Name	Data Type	Description	Required?
Inherits From: `Item`			
`LoginName`	`String (70)` (n-v pair)	The login name for an Unified CCDM user. Additionally, for ISE-enabled users,and all users in installations that use SSO, the following applies. The login name must correspond to an existing Windows Active Directory user,and must be formatted as **<username>@<domain-name>**, where **<username>** is the Windows username and **<domain-name>** is the fully qualified Windows domain name. An example is **user1@testdomain.local**. The login name must exactly match the details in the corresponding Active Directory entry.	Yes
`PassPhrase`	`String (100)` (n-v pair)	The passphrase for the normal user. This is not required when operating with an external security provider.	Depends on ICM configuration
`NewPassPhrase`	`String (100)` (n-v pair)	The new passphrase for the user. Only required when the passphrase is being changed.	No
`Description`	`String (500)` (n-v pair)	The description of this instance of the resource.	No
`FailedAttempts`	`Int` (n-v pair)	This the number of failed login attempts by the user. This is a read-only field only set on Retrieve or Search operations.	No
`AccountLocked`	`Bool` (n-v pair)	A flag indicating whether the user's account has been locked or not. This is a read-only field only set on Retrieve or Search operations. A value of `True` indicates the account is locked. The default is `False`.	No
`PassPhraseExpired`	`Bool` (n-v pair)	A flag indicating whether the user's passphrase has expired or not. Value of true indicated pass phrase expired. Default is `false`.	No
`PassPhraseNeverExpires`	`Bool` (n-v pair)	A flag indicating whether the user's passphrase will never expire. A value of true indicates that the passphrase never expires. Default is `false`.	No

Element Name	Data Type	Description	Required?
PassPhraseChangeEnabled	Bool (n-v pair)	A flag indicating whether the user can change their passphrase or not. A value of true indicates that the user can change their passphrase. Default is true.	No
CreateHomeFolder	Bool (n-v pair)	Used on create only, to indicate whether a home folder should be created automatically for this user. If this is true, a home folder for the user will be created directly below the folder where the user is being created. The folder name will be the same as the user name, followed by the user folder suffix, as set in the Web UI, using **Settings > Security > User Folder Suffix**.	No
HomeFolder	String (n-v pair)	The folder where the user is taken to when they log in. Basic users will be unable to navigate out of this folder. This field is optional, but if specified, the folder must already exist. On create, this value is only used if CreateHomeFolder is not true. On create, if this is not set and CreateHomeFolder is not true, the user's home folder will be set to the folder where the user is being created. On edit, this specifies the new folder to use as the user's home folder. ***Note.*** *When creating a user, do not set this field if you have also set CreateHomeFolder to true.*	No
FirstName	String (36) (n-v pair)	The first name of the user.	No
LastName	String (50) (n-v pair)	The last name of the user.	No
EMail	String (50) (n-v pair)	The email address associated with the user. On some external security providers, this acts as an alternative primary key.	No

Element Name	Data Type	Description	Required?
Expert	String (n-v pair)	A flag indicating when the user has expert mode enabled. A value of 1 indicates that expert mode is enabled.	No
TimeZone	String (n-v pair)	The time-zone in which the user is located. See section 5.6.3 "Dialed Number From Unallocated folder" for the list of valid time zones.	No

Note
The User resource does not use the following Item elements: InternalName, MappedClusterResources and MappedClusterResourceDetails.

4.4.3.3 Associated Name Fields for Users

As well as the fields above, the following associated name fields can also be used to sort User resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.29 Associated Name Fields for Sorting Users

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId

4.4.3.4 Model

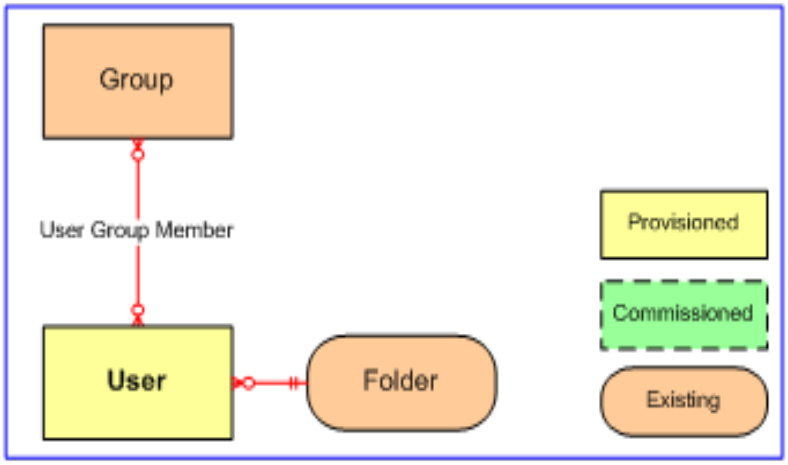


Figure 4.6 User Relationships

4.4.3.5 REST Protocol

Folder REST API Summary				
Base URL	https://<server>:8085/resourcemanagement/rest/resources			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL] /folders/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Sync.	POST	200 OK	[Base URL]/users
Delete	Sync.	DELETE	202 Accepted	[Base URL]/users/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/user
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/users/<id>,<id>

Folder REST API Summary				
Save	Sync.	POST	200 OK	[Base URL]/members/users
Search	Sync.	GET	200 OK	[Base URL]? queryString= type%3aIT_USER&max%3a10
Update	Sync.	PUT	202 Accepted	[Base URL]/users/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/users			

4.5 Provisionable Remote Resource Types

4.5.1 Agent Resource Item

4.5.1.1 Description

The `Agent` resource represents a customer service representative on a specific equipment instance (Peripheral) and contains the information for that agent specific to that equipment. A particular customer service representative may correspond to more than one agent resource. For example, John Smith the person has an agent record for a Voice Peripheral on which he answers phone calls and an agent record on an E-Mail Peripheral on which he answers emails.

4.5.1.2 Fields

The following fields can be read or set for Agent resource objects.

Table 4.30 Agent Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionItem`				
`PeripheralUrn`	`String (10)` (n-v pair)	The peripheral or equipment identifier on which the agent will be located. If -1 is supplied then Unified CCDM will pick the Peripheral automatically. Note: cannot be updated after the Create. Default: -1	Yes	No
`AgentDesktopUrn`	`String (10)` (n-v pair)	The desktop profile to be used by this agent. Default to -1 if not set. Note: can be modified using the Update call. Default: -1	No	No
`PersonUrn`	`String (10)` (n-v pair)	The Person resource to which this agent will be associated. A Person can only have one Agent representation on any one Peripheral but can have a representation on all the relevant underlying Peripherals. For example. a Person can have an Agent Resource on a Voice Peripheral and an Agent Resource on an E-Mail Peripheral. Default: -1	Yes	No

Element Name	Data Type	Description	Required?	Pkey?
PeripheralNumber	String (50) (n-v pair)	This is the agents login ID at the switch.	No	Yes
PeripheralName	String (50) (n-v pair)	This is the name of the agent as known to the peripheral.	No	Yes
Supervisor	Bool (n-v pair)	This is a flag indicating whether the agent can be a supervisor of agent teams or not. Default: false.	No	Yes
AgentStateTrace	Bool (n-v pair)	This is a flag indicating whether the agent state trace data needs to be collected or not. Default: false.	No	Yes
DomainName	String (255) (n-v pair)	This is the DNS format of the active directory domain name. This is populated only if the agent is set as a Supervisor	No	Yes
DomainLoginName	String (64) (n-v pair)	This is the domain login name. This is populated only if the agent is set as a supervisor.	No	Yes
DomainUserName	String (255) (n-v pair)	This is the user name. This is populated only if the agent is set as a supervisor.	No	Yes
DomainUserGuid	String (36) (n-v pair)	This is the unique identifier of the user in the active directory. This is populated only if the agent is set as a supervisor.	No	Yes
DomainPassPhrase	String (50) (n-v pair)	For future use. This is the pass phrase for the user on the Active Directory	No	Yes
UserDeletable	Bool (n-v pair)	Indicates if the item can be deleted by end users as opposed to the Unified CCDM background processes. Note: used for Unified CCE parent/child configurations only.	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
UserGroupId	String (n-v pair)	The User Group ID for this Agent on Unified CCE.	No	Yes
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.1.3 Associated Name Fields for Agents

As well as the fields above, the following associated name fields can also be used to sort Agent resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.31 Associated Name Fields for Sorting Agents

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentAgentDesktopDisplayName	Agent.AgentDesktopUrn
ParentDepartmentDisplayName	Agent.DepartmentBizUrn
ParentPeripheralDisplayName	Agent.PeripheralUrn
ParentPersonDisplayName	Agent.PersonUrn
ParentPersonFirstName	Agent.PersonUrn
ParentPersonLastName	Agent.PersonUrn
ParentPersonLoginName	Agent.PersonUrn
ParentPersonLoginEnabled	Agent.PersonUrn

4.5.1.4 Model

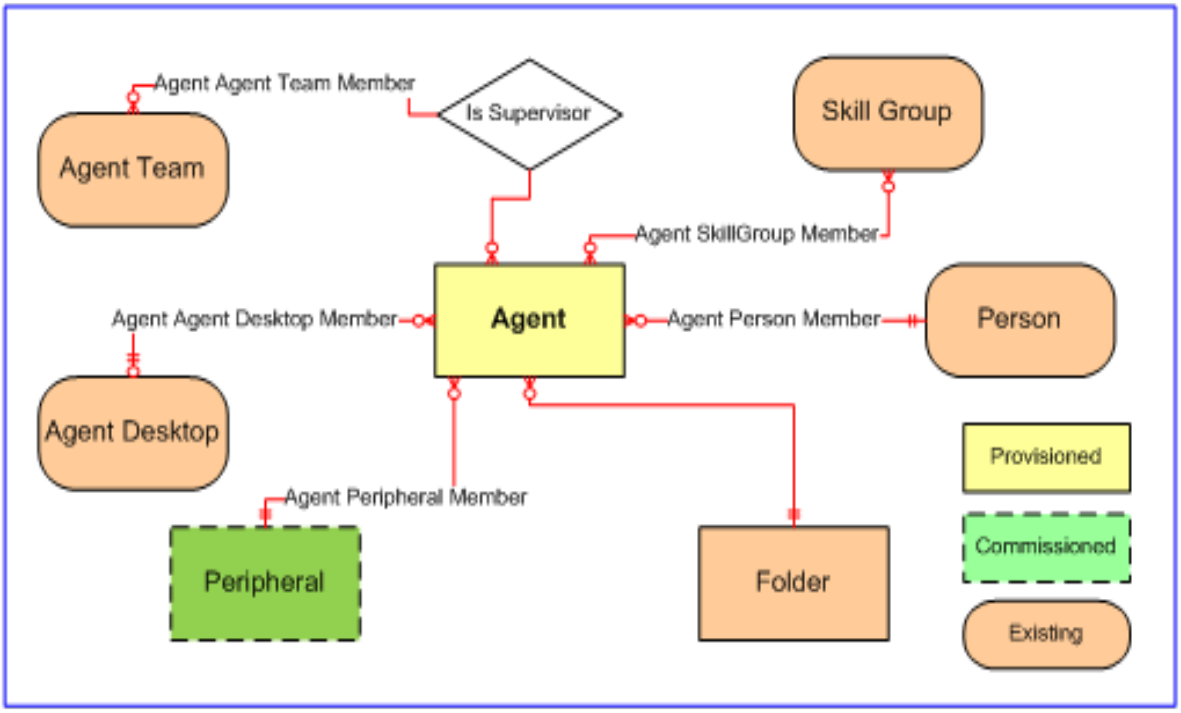


Figure 4.7 Agent Relationships

4.5.1.5 REST Protocol

Agent REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL] /agents/<id>,<id>/audits? fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/agents`

Agent REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL]/agents/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/agent
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/agents/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/agents
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_AGENT&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/agents/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/agents			

4.5.2 Agent Desktop Resource Item

4.5.2.1 Description

The AgentDesktop resource represents the options available to an agent, typically via their CTI desktop.

This resource has an optional resource membership to a dialed number which represents the Ring No Answer Dialed Number.

4.5.2.2 Fields

The following fields can be read or set for Agent Desktop resources.

Table 4.32 Agent Desktop Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
AvailableAfterIncoming	Bool (n-v pair)	A flag indicating whether to consider an agent to be available after handling an incoming call.	No	Yes
AvailableAfterOutgoing	Bool (n-v pair)	A flag indicating whether to consider an agent to be available after handling an outbound call.	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
WrapupDataIncomingMode	Int (n-v pair)	A flag indicating whether the agent is allowed or required to enter wrap-up data after handling an incoming call. This takes the values 0: Required; 1: Optional; 2: Not allowed.	No	Yes
WrapupDataOutgoingMode	Int (n-v pair)	A flag indicating whether the agent is allowed or required to enter wrap-up data after handling an outbound call. This takes the values 0: Required; 1: Optional; 2: Not allowed.	No	Yes
AutoAnswerEnabled	Bool (n-v pair)	A flag indicating whether the calls to the agent are automatically answered.	No	Yes
IdleReasonRequired	Bool (n-v pair)	A flag indicating whether the agent should enter a reason before entering the Idle state.	No	Yes
LogoutNonActivityTime	Int (n-v pair)	The number of seconds after which the agent will be logged out automatically if there is no activity at the agent desktop. This value is between 10 and 7200.	No	Yes
LogoutReasonRequired	Bool (n-v pair)	A flag indicating whether the agent is required to enter a reason before logging out.	No	Yes
SupervisorCallsAllowed	Bool (n-v pair)	A flag indicating whether the agent can initiate supervisor assisted calls.	No	Yes
SupervisorAssistCall Method	Int (n-v pair)	A value that indicates whether system would create a consultative or a blind conference call for the supervisor assistance request. This takes the values 0: Consultative; 1: Blind conference.	No	Yes
EmergencyCallMethod	Int (n-v pair)	A value that indicates whether system would create a consultative or a blind conference call for the emergency call request. This takes the values 0: Consultative; 1: Blind conference.	No	Yes
AutoRecordOnEmergency	Bool (n-v pair)	A flag indicating whether to automatically record or not to record when an emergency call request is started.	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
AgentToAgentCalls Allowed	Bool (n-v pair)	A flag indicating whether calls to other agents are allowed.	No	Yes
OutboundAccess International	Bool (n-v pair)	A flag indicating whether the agent can initiate international calls.	No	Yes
OutboundAccessPublic Network	Bool (n-v pair)	A flag indicating whether the agent can initiate calls through public network.	Yes	Yes
OutboundAccessPrivate Network	Bool (n-v pair)	A flag indicating whether the agent can initiate calls through private network.	Yes	Yes
OutboundAccessOperator Assisted	Bool (n-v pair)	A flag indicating whether the agent can initiate operator assisted calls.	No	Yes
OutboundAccessPBX	Bool (n-v pair)	A flag indicating whether the agent can initiate outbound PBX calls.	No	Yes
NonACDCallsAllowed	Bool (n-v pair)	A flag indicating whether the agent can place or handle non-ACD calls.	No	Yes
QualityRecordingRate	Int (n-v pair)	A value in seconds indicating how frequently calls to the agent are recorded.	No	Yes
RecordingMode	Int (n-v pair)	A value that indicates whether the call requests are routed through Unified Communications Manager. Value 0 indicates that the calls do not get routed through Unified Communications Manager.	No	Yes
WorkModeTimer	Int (n-v pair)	A value indicating the wrap-up time out in seconds. This value is between 1 and 7200.	No	Yes
RingNoAnswerTime	Int (n-v pair)	A value indicating the number of seconds a call may ring at the agent's station before it is redirected. This value is between 1 and 120.	No	Yes
SilentMonitorWarning Message	Bool (n-v pair)	A flag indicating whether a warning message will appear on the agent's desktop when silent monitoring is started.	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
SilentMonitorAudible Indication	Bool (n-v pair)	A flag indicating whether an audio click will sound when silent monitoring is started.	No	Yes
DefaultDevicePort Address	String (n-v pair)	A value to override the default port address of the agent's telephony device.	No	Yes
AgentCanSelectGroup	Bool (n-v pair)	A flag indicating whether the agent can select which group they are logged in to.	No	Yes
RemoteAgentType	Int (n-v pair)	A value indicating how mobile agents who use this dial plan are handled. This takes the values 0: Local agent, no remote access; 1: Use call by call mobile agent routing; 2: Use nailed connection for mobile agents; 3: Agent chooses at login.	No	Yes
RemoteLoginWithout Desktop	Bool (n-v pair)	A flag indicating whether mobile agents are allowed to login without a desktop.	No	Yes
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.2.3 Associated Name Fields for Agent Desktops

As well as the fields above, the following associated name fields can also be used to sort Agent Desktop resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.33 Associated Name Fields for Sorting Agent Desktops

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	AgentDesktop.DepartmentBizUrn

4.5.2.4 Model

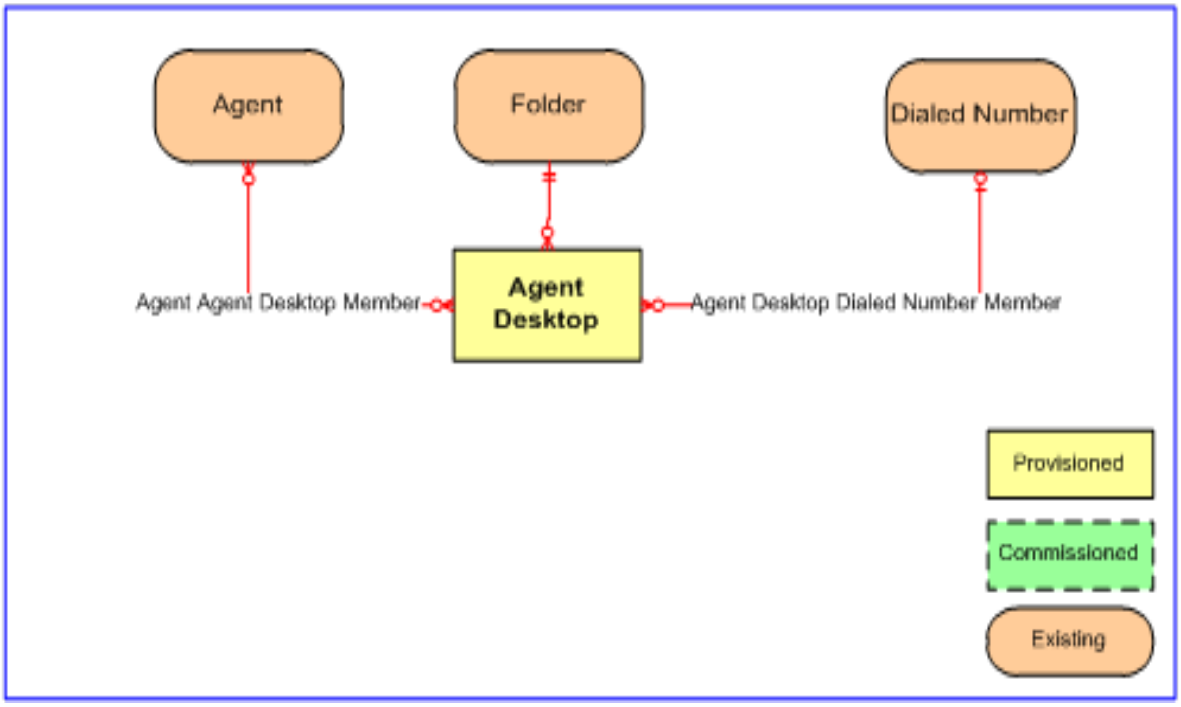


Figure 4.8 Agent Desktop Relationships

4.5.2.5 REST Protocol

Agent REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL]/agent-desktops/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/agent-desktops`

Agent REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL]/agent-desktops/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/agent-desktop
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/agent-desktops/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/agent-desktops
Search	Sync.	GET	200 OK	[Base URL]? queryString=type%3aIT_AGENT_DESKTOP&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/agent-desktops/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/agent-desktops			

4.5.3 Agent Team Resource Item

4.5.3.1 Description

The `AgentTeam` resource represents a specific collection of agent resources, usually with an assigned supervisor. Call routing does not use this resource type. It is used for line management and performance management. The Agent team is linked to a specific equipment peripheral to ensure that agents on an equipment instance are grouped together. For other grouping types use a folder resource or hierarchy. For the Cisco UCCE call routing platform an agent is typically only a member of one agent team.

4.5.3.2 Fields

The following fields can be read or set for Agent Team resources:

Table 4.34 Agent Team Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
PeripheralUrn	String (10) (n-v pair)	The peripheral or equipment identifier on which the agent team will be located. If -1 is supplied then Unified CCDM will pick the Peripheral automatically. Note: cannot be updated after the Create.	Yes	No
DialedNumberUrn	String (10) (n-v pair)	The default dialed number associated with this agent team.	No	No
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.3.3 Associated Name Fields for Agent Teams

As well as the fields above, the following associated name fields can also be used to sort Agent Team resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.35 Associated Name Fields for Sorting Agent Teams

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	AgentTeam.DepartmentBizUrn
ParentDialedNumberDisplayName	AgentTeam.DialedNumberUrn
ParentPeripheralDisplayName	AgentTeam.PeripheralUrn

4.5.3.4 Model

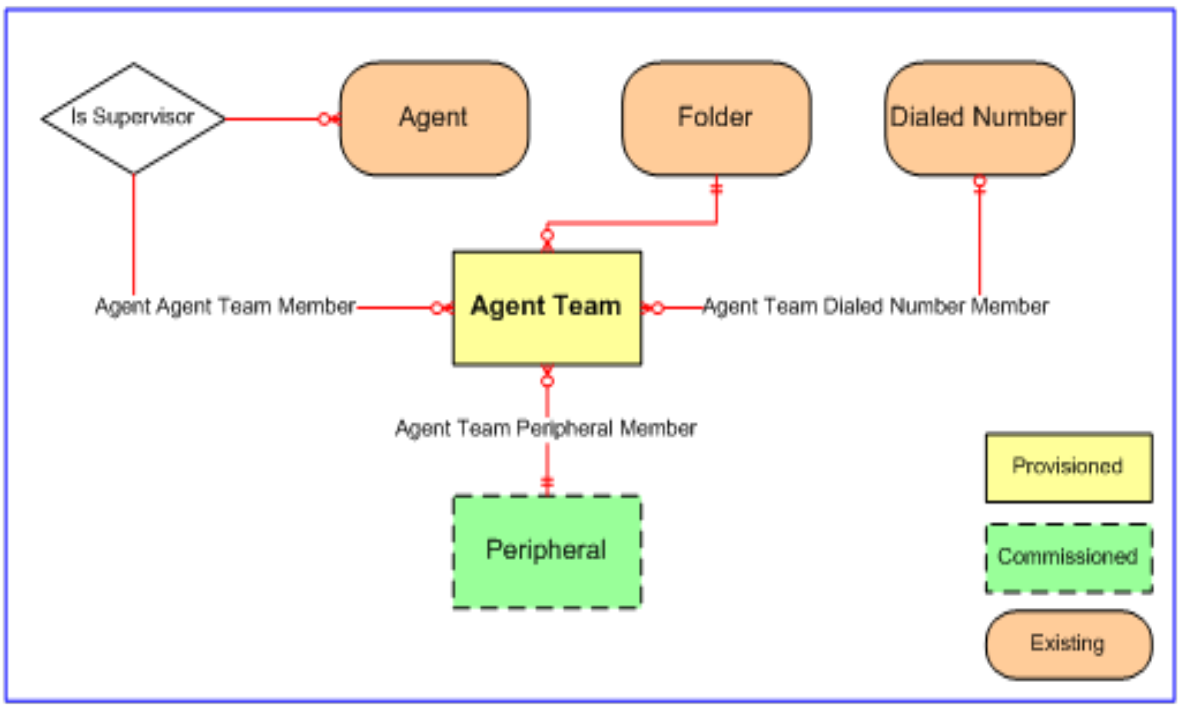


Figure 4.9 Agent Team Relationships

4.5.3.5 REST Protocol

Agent Team REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL]/agent-teams/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/agent-teams`

Agent Team REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL]/agent-teams/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/agent-team
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/agent-teams/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/agent-teams
Search	Sync.	GET	200 OK	[Base URL]? queryString= type%3aIT_AGENT_TEAM&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/agent-teams/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/agent-teams			

4.5.4 Call Type Resource Item

4.5.4.1 Description

The CallType resource represents a classifier applied to a specific class of calls. This allows the treatment of calls to be handled by a specific routing script; typically after the call has been segmented by IVR treatment or simple caller enter digits.

4.5.4.2 Fields

The following fields can be read or set for Call Type resources:

Table 4.36 Call Type Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
ServiceLevelThreshold	Int (n-v pair)	The service level threshold for the Call Type. Possible values are: 0: None (default) 1: Use global settings 2 or greater: Threshold value in seconds	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
ServiceLevelType	Short (n-v pair)	This is the type that defines how service levels are calculated for the Call Type. Defaults to 0. This takes the values: 0: Default 1: Ignore abandoned calls 2: Abandoned calls have negative impact 3: Abandoned calls have positive impact.	No	Yes
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.4.3 Associated Name Fields for Call Types

As well as the fields above, the following associated name fields can also be used to sort Call Type resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.37 Associated Name Fields for Sorting Call Types

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	CallType.DepartmentBizUrn

4.5.4.4 Model

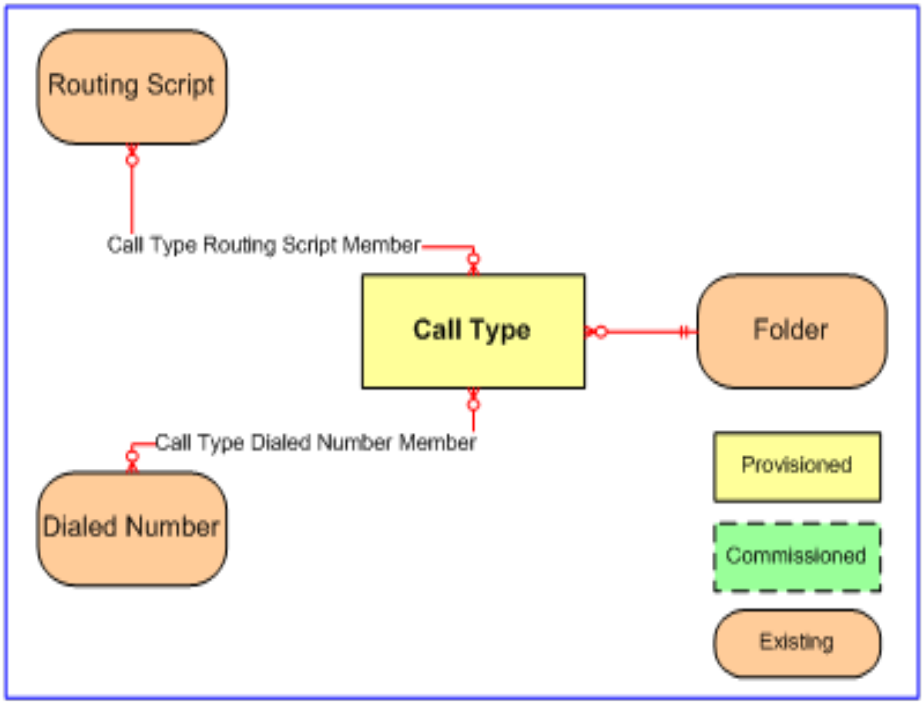


Figure 4.10 Call Type Relationships

4.5.4.5 REST Protocol

Call Type REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL]/call-types/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/call-types`

Call Type REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL]/call-types/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/call-type
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/call-types/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/call-types
Search	Sync.	GET	200 OK	[Base URL]? queryString= type%3aIT_CALL_TYPE&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/call-types/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/call-types			

4.5.5 Calling Search Space Resource Item

4.5.5.1 Description

The CallingSearchSpace resource represents a specific collection of Route Partitions that are considered before an IP call can be made.

4.5.5.2 Fields

The following fields can be read or set for Calling Search Space resources:

Table 4.38 Calling Search Space Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
Clause	String(500) (n-v pair)	CSV list of Route Partition Ids	Yes	Yes

4.5.5.3 Associated Name Fields for Calling Search Spaces

As well as the fields above, the following associated name fields can also be used to sort Calling Search Space resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.39 Associated Name Fields for Sorting Calling Search Spaces

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId

4.5.5.4 Model

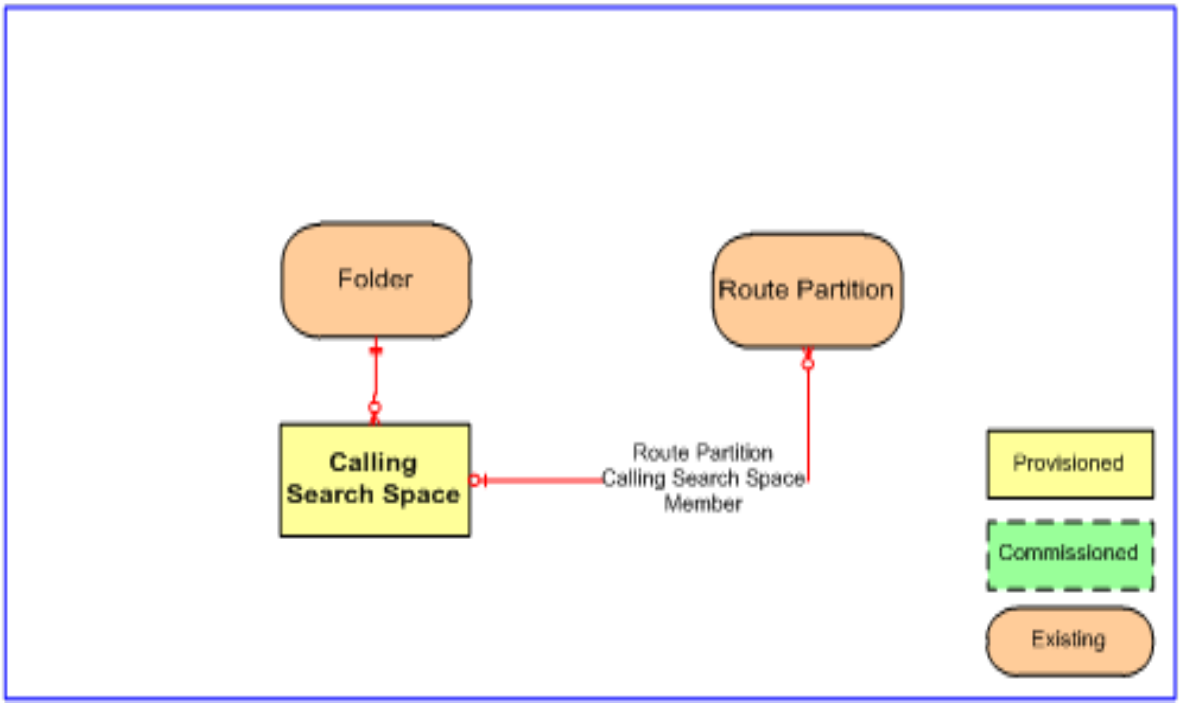


Figure 4.11 Calling Search Space Relationships

4.5.5.5 REST Protocol

Calling Search Space REST API Summary				
Base URL	https://<server>:8085/resourcemanagement/rest/resources			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL]/calling-search-spaces/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/calling-search-spaces
Delete	Async.	DELETE	202 Accepted	[Base URL]/calling-search-spaces/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/calling-search-space
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/calling-search-spaces/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/calling-search-spaces/
Search	Sync.	GET	200 OK	[Base URL]? queryString=type%3aIT_CALLLING_SEARCH_SPACE&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/calling-search-spaces/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/calling-search-spaces			

4.5.6 Department Resource Item

4.5.6.1 Description

The Department resource is a way of grouping resources according to a business need.

4.5.6.2 Fields

The following fields can be read or set for Department resources:

Table 4.40 Department Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionItem`				
No further fields.				

4.5.6.3 Associated Name Fields for Departments

As well as the fields above, the following associated name fields can also be used to sort Department resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.41 Associated Name Fields for Departments

Associated Name Field	Related Element
`ParentCreatedByLoginName`	`Entry.CreatedById`
`ParentFolderPath`	`Item.FolderId`
`ParentOwnerName`	`DimensionItem.OwnerId`

4.5.6.4 Model

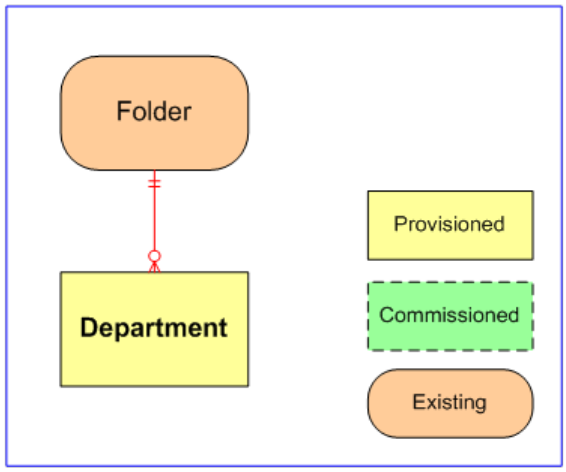


Figure 4.12 Department Relationships

4.5.6.5 REST Protocol

Department REST API Summary				
Base URL	https://<server>:8085/resourcemanagement/rest/resources			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL]/department/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/departments
Delete	Async.	DELETE	202 Accepted	[Base URL]/departments/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/department
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>

Department REST API Summary				
Retrieve	Sync.	GET	200 OK	[Base URL]/departments/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/departments
Search	Sync.	GET	200 OK	[Base URL]? queryString= type%3aIT_DEPARTMENT&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/departments/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/departments			

4.5.7 Dialed Number Resource Item

4.5.7.1 Description

The DialedNumber resource represents the entry point into the contact center product set. The call can be pre-screened and segmented using IVR or go straight through to an agent queuing solution or a combination of both. Care is needed with this resource type since many organizations have only a single 01800 dialed number entry point and invalid removal of this can make many agents idle.

4.5.7.2 Fields

The following fields can be read or set for Dialed Number resources:

Table 4.42 Dialed Number Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
RoutingClientUrn	String (10) (n-v pair)	The routing client that services this dialed number.	Yes	No
MediaRoutingDomainUrn	String (10) (n-v pair)	The media routing domain that services this dialed number.	Yes	No

Element Name	Data Type	Description	Required?	Pkey?
DialedNumber	String (32) (n-v pair)	This is a string indicating the dialable number for the directory number. This can be the actual numeric number or a route pattern with wildcards.	No	No
PermitApplicationRouting	Bool (n-v pair)	Flag determining whether application routing is allowed on this dialed number.	No	Yes
SelfCareEnabled	Bool (n-v pair)	Flag indicating whether self-care is enabled for this dialed number.	No	No
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.7.3 Associated Name Fields for Dialed Numbers

As well as the fields above, the following associated name fields can also be used to sort Dialed Number resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.43 Associated Name Fields for Sorting Dialed Numbers

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	DialedNumber.DepartmentBizUrn

4.5.7.4 Model

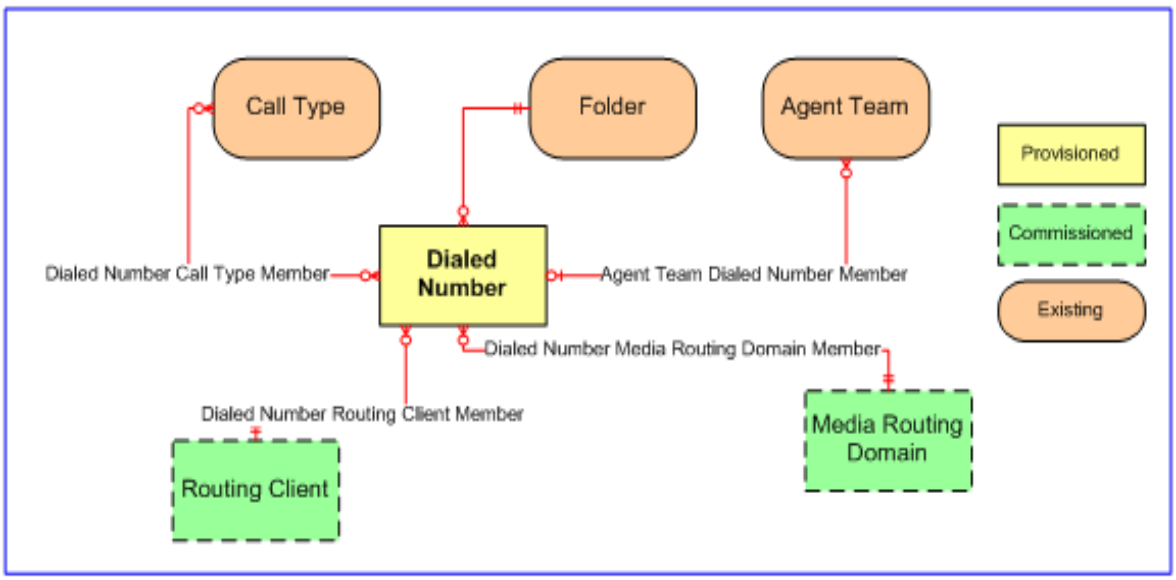


Figure 4.13 Dialed Number Relationships

4.5.7.5 REST Protocol

Dialed Number REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL]/dialed-numbers/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/dialed-numbers`
Delete	Async.	`DELETE`	`202 Accepted`	`[Base URL]/dialed-numbers/<id>,<id>`
Describe	Sync.	`GET`	`200 OK`	`[Base URL]/meta/dialed-number`

Dialed Number REST API Summary				
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/dialed-numbers/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/dialed-numbers
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_DIALED_NUMBER&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/dialed-numbers/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/dialed-numbers			

4.5.8 Enterprise Skillgroup Resource Item

4.5.8.1 Description

The EnterpriseSkillgroup represents a specific collection of skillgroups.

4.5.8.2 Fields

The following fields can be read or set for Enterprise Skillgroup resources:

Table 4.44 Enterprise Skill Group Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.8.3 Associated Name Fields for Enterprise Skillgroups

As well as the fields above, the following associated name fields can also be used to sort Enterprise Skillgroup resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.45 Associated Name Fields for Sorting Enterprise Skillgroups

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	EnterpriseSkillGroup.DepartmentBizUrn

4.5.8.4 Model

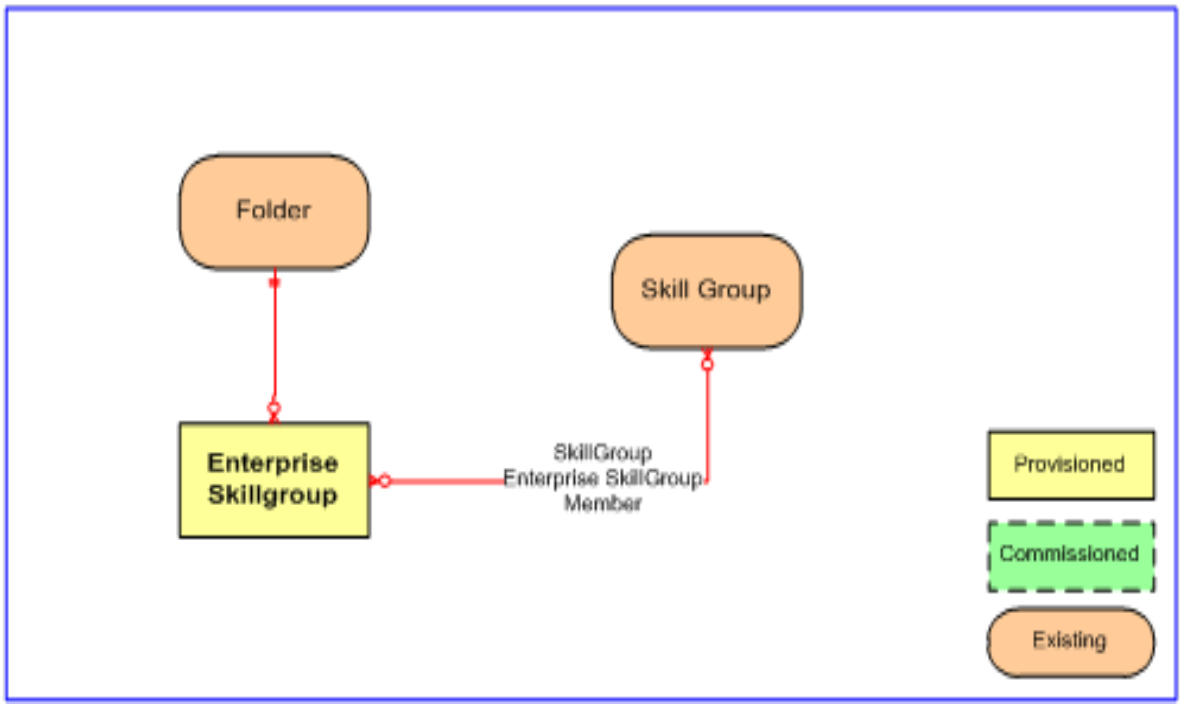


Figure 4.14 Enterprise Skillgroup Relationships

4.5.8.5 REST Protocol

Enterprise Skillgroup REST API Summary				
Base URL	https://<server>:8085/resourcemanagement/rest/resources			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL]/enterprise-skillgroups/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/enterprise-skillgroups
Delete	Async.	DELETE	202 Accepted	[Base URL]/enterprise-skillgroups/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/enterprise-skillgroup
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/enterprise-skillgroups/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/enterprise-skillgroups/
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_ENTERPRISE_SKILLGROUP&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/enterprise-skillgroups/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/enterprise-skillgroups			

4.5.9 Expanded Call Variable Resource Item

4.5.9.1 Description

The `ExpandedCallVariable` resource is used to hold scripting and other data variables.

4.5.9.2 Fields

The following fields can be read or set for Expanded Call Variable resources:

Table 4.46 Expanded Call Variable Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionItem`				
`ECCArray`	`Bool` (n-v pair)	This is a flag indicating whether the call variable is an array. Default: `false`.	No	Yes
`VariableEnabled`	`Bool` (n-v pair)	This is a flag indicating whether the call variable is enabled. Default: `false`.	No	Yes
`GeoTelProvided`	`Bool` (n-v pair)	This is a flag indicating whether the call variable is provided by Cisco. Default: `false`.	No	Yes
`MaximumArraySize`	`Short` (n-v pair)	This value indicates the number of elements in the array if the call variable is set as an array. The values are between 1 and 255. Default: 0	No	Yes
`MaximumLength`	`Short` (n-v pair)	This value indicates the maximum length for the call variable. This value is between 1 and 210. Default: 1	No	Yes
`Persistent`	`Bool` (n-v pair)	This is a flag indicating whether the call variable is persistent or not. Default: `true`.	No	Yes

4.5.9.3 Associated Name Fields for Expanded Call Variables

As well as the fields above, the following associated name fields can also be used to sort Expanded Call Variable resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.47 Associated Name Fields for Sorting Expanded Call Variables

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId

4.5.9.4 Model

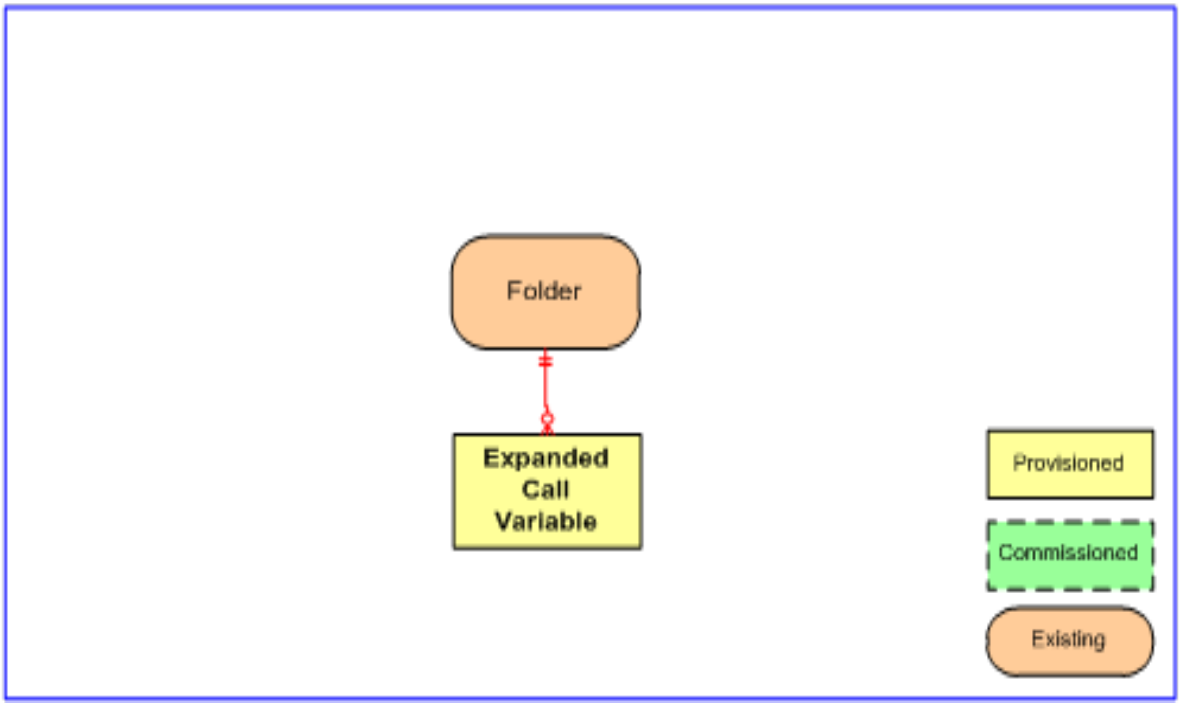


Figure 4.15 Expanded Call Variable Relationships

4.5.9.5 REST Protocol

Expanded Call Variable REST API Summary	
Base URL	https://<server>:8085/resourcemanagement/rest/resources
IO Format	XML or JSON

Expanded Call Variable REST API Summary				
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL]/expanded-call-variables/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/expanded-call-variables
Delete	Async.	DELETE	202 Accepted	[Base URL]/expanded-call-variables/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/label
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/expanded-call-variables/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/expanded-call-variables
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_EXPANDED_CALL_VARIABLE&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/expanded-call-variables/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/expanded-call-variables			

4.5.10 IVR Script Resource Item

4.5.10.1 Description

The `IVRScript` resource represents the IVR script on the Customer Voice Portal (CVP). It is used to control call routing. IVR scripts can be created using Cisco Unified Call Studio.

Note. *In the Unified CCDM web application, IVR scripts are called IVR apps.*

4.5.10.2 Fields

The following fields can be read or set on IVR script resource objects.

Table 4.48 IVR Script Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionItem`				
`ScriptName`	`String (100)` (n-v pair)	The name of the IVR script on the IVR app server.	Yes	Yes
`Description`	`String (255)` (n-v pair)	The description of the IVR script on the IVR app server.	No	Yes

4.5.10.3 Associated Name Fields for IVR Scripts

As well as the fields above, the following associated name fields can also be used to sort IVR script resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.49 Associated Name Fields for Sorting IVR Script

Associated Name Field	Related Element
`ParentCreatedByLoginName`	`Entry.CreatedById`
`ParentFolderPath`	`Item.FolderId`
`ParentOwnerName`	`DimensionItem.OwnerId`

4.5.10.4 Model

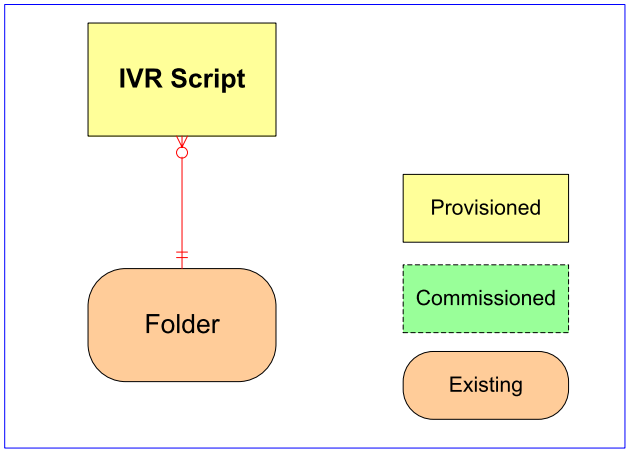


Figure 4.16 IVR Script Relationships

4.5.10.5 REST Protocol

IVR Scripts REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL]/ivr-scripts/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…`
Create	Async	[POST]	[202 Accepted]	`[Base URL]/ivr-scripts`
Delete	Async.	`DELETE`	`202 Accepted`	`[Base URL]/ivr-scripts/<id>,<id>`
Describe	Sync.	`GET`	`200 OK`	`[Base URL]/meta/ivr-script`
Move	Sync.	`PUT`	`200 OK`	`[Base URL]/<destinationid>`

IVR Scripts REST API Summary				
Retrieve	Sync.	GET	200 OK	[Base URL]/ivr-scripts/<id>,<id>
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_IVR_SCRIPT&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/ivr-scripts/<id>,<id>
Deploy	Async.	[POST]	[202 Accepted]	Base URL/ivr-scripts/deploy?contentType=application/zip
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/ivr-scripts			

4.5.11 Label Resource Item

4.5.11.1 Description

The Label resource represents a string that is passed to a routing client for each network target.

4.5.11.2 Fields

The following fields can be read or set for Label resources:

Table 4.50 Label Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
RoutingClientUrn	String (n-v pair)	Parent resource that must be supplied during the create process.	Yes	No
LabelType	Short (n-v pair)	This value indicates the type of the label. This takes the values 0: Normal; 1: DNIS override; 2: Busy; 3: Ring; 4: Post-Query; 5: Resource.	Yes	Yes

4.5.11.3 Associated Name Fields for Labels

As well as the fields above, the following associated name fields can also be used to sort Label resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.51 Associated Name Fields for Sorting Labels

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentRoutingClientDisplayName	Label.RoutingClientUrn

4.5.11.4 Model

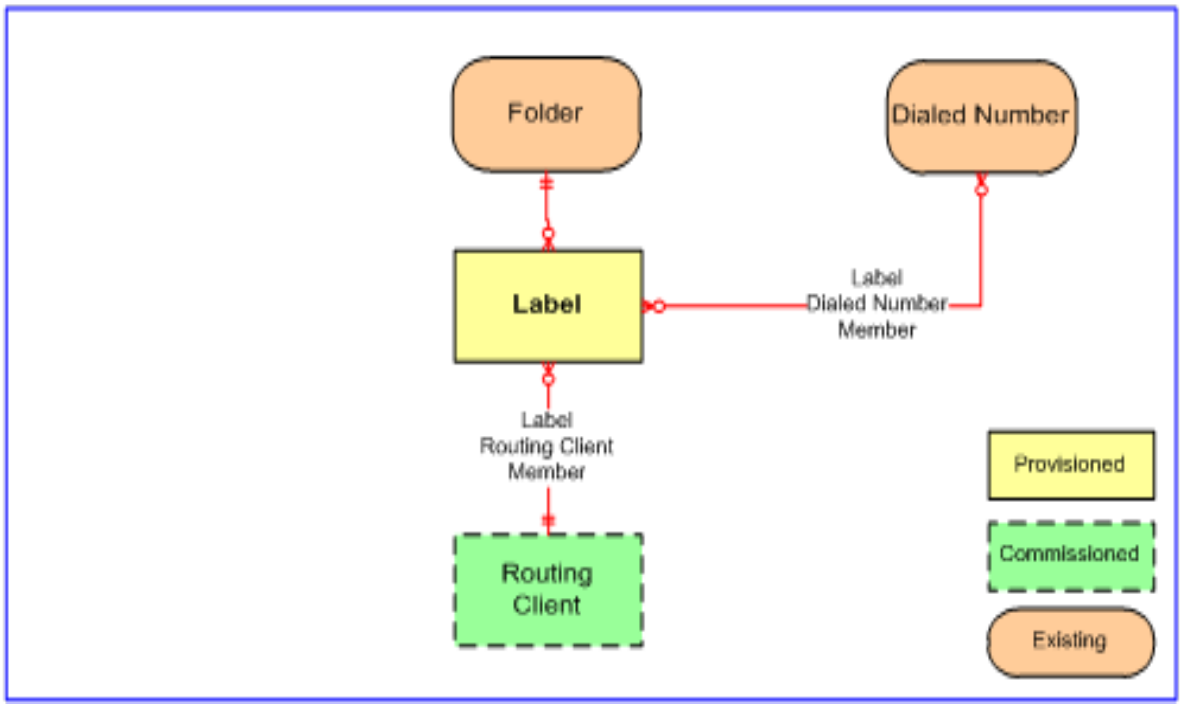


Figure 4.17 Label Relationships

4.5.11.5 REST Protocol

Label REST API Summary				
Base URL	https://<server>:8085/resourcemanagement/rest/resources			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL] /labels/<id>,<id>/audits? fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/labels
Delete	Async.	DELETE	202 Accepted	[Base URL]/labels/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/label
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/labels/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/labels
Search	Sync.	GET	200 OK	[Base URL]?queryString= type%3aIT_LABEL&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/labels/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/labels			

4.5.12 Media File Resource Item

4.5.12.1 Description

The MediaFile resource represents a media file on CVP servers. This is a .wav file used by IVR to play back prompts and messages to callers.

4.5.12.2 Fields

The following fields can be read or set on media file resource objects.

Table 4.52 Media File Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
Filename		The name of the media file on the media file server.	Yes	Yes
Path		The file path of the media file on the media file server. This field cannot be edited.	Yes	Yes
Size		The size of the item. This field cannot be created or edited.	N/A	Yes
Checksum		The check-sum of the item. This field cannot be created or edited.	N/A	Yes

4.5.12.3 Associated Name Fields for Media Files

As well as the fields above, the following associated name fields can also be used to sort media file resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.53 Associated Name Fields for Sorting Media Files

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId

4.5.12.4 Model

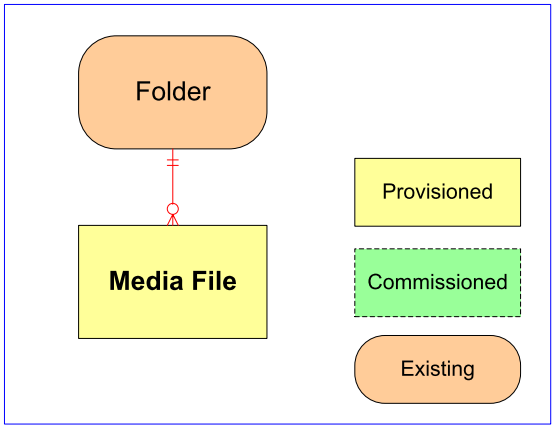


Figure 4.18 Media File Relationships

4.5.12.5 REST Protocol

Media files REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL]/mediafiles/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/mediafiles`
Delete	Async.	`DELETE`	`202 Accepted`	`[Base URL]/mediafiles/<id>,<id>`
Describe	Sync.	`GET`	`200 OK`	`[Base URL]/meta/mediafile`
Move	Sync.	`PUT`	`200 OK`	`[Base URL]/<destinationid>`
Retrieve	Sync.	`GET`	`200 OK`	`[Base URL]/mediafiles/<id>,<id>`
Save	Sync.	`POST`	`200 OK`	`[Base URL]/members/mediafiles`

Media files REST API Summary				
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_MEDIAFILE&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/mediafiles/<id>,<id>
Upload	Async	PUT	[200 OK]	Base URL/mediafiles/3565/content?contentType=audio/wav
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/mediafiles			

4.5.13 Network VRU Script Resource Item

4.5.13.1 Description

The NetworkVruScript resource represents a script run on a Network VRU to handle a call.

4.5.13.2 Fields

The following fields can be read or set for Network VRU Script resources:

Table 4.54 Network VRU Script Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
VruScriptName	String (n-v pair)	This is the name of the script on the VRU.	Yes	Yes
NetworkVruUrn	String (10) (n-v pair)	Foreign key to the Network VRU that this script is associated with.	Yes	No
Interruptible	Bool (n-v pair)	This is a flag indicating whether the script can be interrupted. Default: False.	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
Overridable	Bool (n-v pair)	This is flag indicating whether the VRU script can override its interruptible flag. Default: False.	No	Yes
ConfigParam	String (n-v pair)	This is string that is sent to the VRU for initialization.	No	Yes
Timeout	Int (n-v pair)	This is the number of seconds for which the software will wait for a response from the routing client after directing it to run the script. Default: 0.	Yes	Yes
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.13.3 Associated Name Fields for Network VRU Scripts

As well as the fields above, the following associated name fields can also be used to sort Network VRU Script resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.55 Associated Name Fields for Sorting Network VRU Scripts

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	NetworkVruScript.DepartmentBizUrn
ParentNetworkVruDisplayName	NetworkVruScript.NetworkVruUrn

4.5.13.4 Model

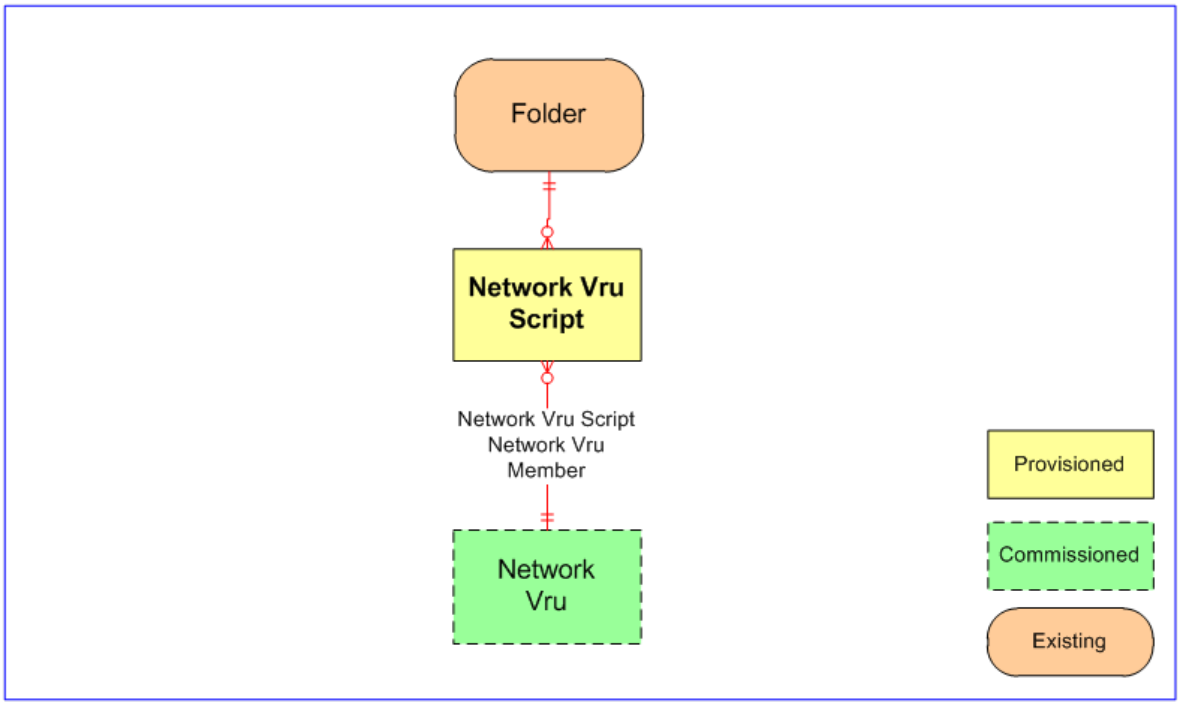


Figure 4.19 Network VRU Script Relationships

4.5.13.5 REST Protocol

Network VRU Script REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL]/network-vru-scripts/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/network-vru-scripts`

Network VRU Script REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL]/network-vru-scripts/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/network-vru-script
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/network-vru-scripts/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/network-vru-scripts
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_NETWORK_VRU_SCRIPT&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/network-vru-scripts/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/network-vru-scripts			

4.5.14 Person Resource Item

4.5.14.1 Description

The `Person` resource represents any person resource on the system, not only customer service representatives but also managers and end customers (people not registered on switch) It is often used for a high speed data dip for CRM-style data-directed routing and IVR services.

4.5.14.2 Fields

The following fields can be read or set for Person resources.

Table 4.56 Fields in Person Resource Item

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
Gender	String (20) (n-v pair)	This is the title for the person. This is not a constrained vocabulary.	No	No
DOB	DateTime (n-v pair)	This is the date of birth of the person.	No	No
DOD	DateTime (n-v pair)	This is the date of death of the person.	No	No
FirstName	String (50) (n-v pair)	This is the first name of the person.	Yes	Yes
LastName	String (50) (n-v pair)	This is the last name of the person.	Yes	Yes
Address	String (255) (n-v pair)	This is the communication address of the person if specified.	No	No
EmployeeType	String (20) (n-v pair)	The tenant specific classification if this person is an employee. This is not a constrained vocabulary.	No	No
GovernmentID	String (50) (n-v pair)	The country specific identifier for this person for example, Social Security number.	No	No
EmployeeID	String (50) (n-v pair)	The company specific identifier for this person for example, payroll number.	No	No
Grade	Short (n-v pair)	The grade or rank of this person within the company.	No	No
Competence	Short (n-v pair)	The competence level of this person within the specified grade.	No	No
UserID	Guid (n-v pair)	The primary identifier of this user	No	No
LoginName	String (50) (n-v pair)	This is the login name associated with the person. This is unique	Yes	Yes

Element Name	Data Type	Description	Required?	Pkey?
PassPhrase	String (50) (n-v pair)	This is the pass phrase used by the person with the associated login name.	Depends on ICM configuration	Yes
LoginEnabled	Bool (n-v pair)	This field indicates whether login for the person is enabled or not.	No	Yes
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.14.3 Associated Name Fields for Persons

As well as the fields above, the following associated name fields can also be used to sort Person resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.57 Associated Name Fields for Sorting Persons

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentUserLoginName	Person.UserID
ParentDepartmentDisplayName	Person.DepartmentBizUrn

4.5.14.4 Model

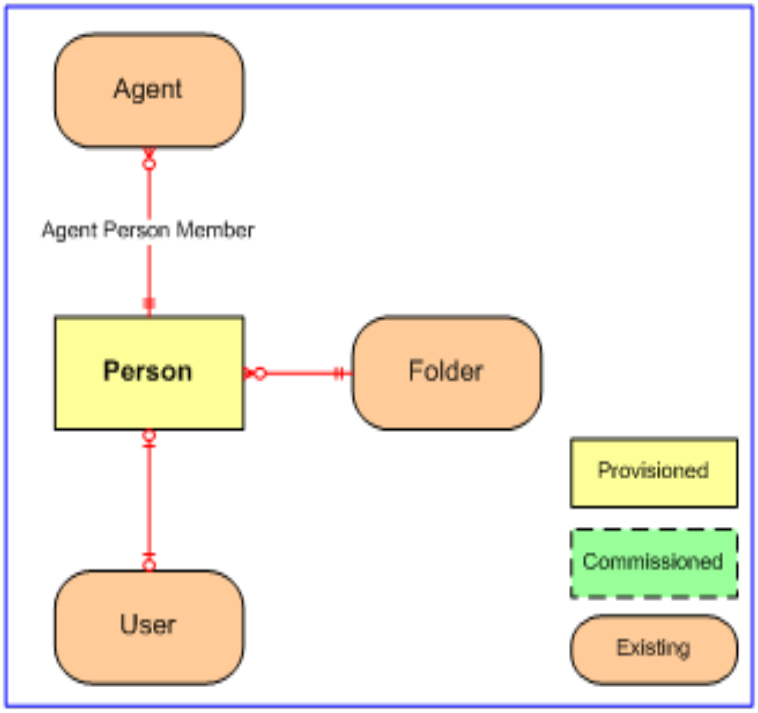


Figure 4.20 Person Relationships

4.5.14.5 REST Protocol

Person REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL] /persons/<id>,<id>/audits? fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/persons`

Person REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL]/persons/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/person
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/persons/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/persons
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_PERSON&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/persons/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/persons			

4.5.15 Precision Attribute Resource Item

4.5.15.1 Description

The PrecisionAttribute resource represents the attributes that may belong to an Agent. It is used for precision queue based call routing.

4.5.15.2 Fields

The following fields can be read or set for Precision Attribute resources:

Table 4.58 Precision Attribute Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
AttributeDataType	Int (n-v pair)	Use this parameter to assign a data type to the attribute using the following values: 3 = Boolean, 4 = Proficiency (special form of Integer)	Yes	Yes
MinimumValue	String (n-v pair)	For Proficiency data types, valid values are 1-10. Otherwise, NULL	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
MaximumValue	String (n-v pair)	For Proficiency data types, valid values are 1-10. Otherwise, NULL	No	Yes
DefaultValue	String (n-v pair)	Use this parameter to specify a default value for the attribute when assigned to an agent if no explicit value is provided. For Boolean data types, valid default values are True and False. For Proficiency data types, valid default values are 1-10	Yes	Yes
AppearsOnDesktop	Bool (n-v pair)	For future use.	No	Yes
SettableByAgent	Bool (n-v pair)	For future use.	No	Yes
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.15.3 Associated Name Fields for Precision Attributes

As well as the fields above, the following associated name fields can also be used to sort Precision Attribute resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.59 Associated Name Fields for Sorting Precision Attributes

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	PrecisionAttribute. DepartmentBizUrn

4.5.15.4 Model

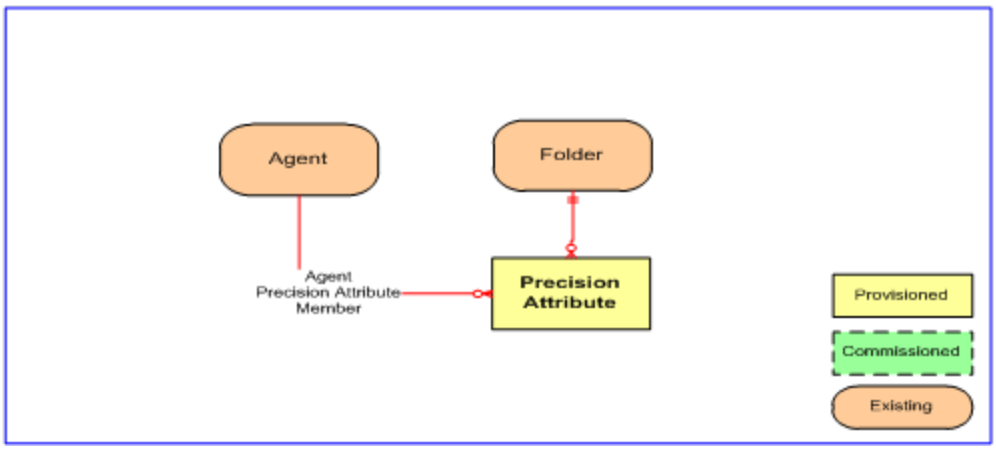


Figure 4.21 Precision Attribute Relationships

4.5.15.5 REST Protocol

Precision Attribute Resource REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/precision-attributes`
Delete	Async.	`DELETE`	`202 Accepted`	`[Base URL]/precision-attributes/<id>,<id>`
Describe	Sync.	`GET`	`200 OK`	`[Base URL]/meta/precision-attribute`
Move	Sync.	`PUT`	`200 OK`	`[Base URL]/<destinationid>`
Retrieve	Sync.	`GET`	`200 OK`	`[Base URL]/precision-attributes/<id>,<id>`
Save	Sync.	`POST`	`200 OK`	`[Base URL]/members/precision-attributes`

Precision Attribute Resource REST API Summary				
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_PRECISION_ATTRIBUTE&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/precision-attributes/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/precision-attributes			

4.5.16 Precision Queue Step Resource Item

4.5.16.1 Description

The `PrecisionQueueStep` resource is a set of ordered steps that apply to a Precision Queue.

Note
A Precision Queue Step cannot exist on its own: it is always associated with a single Precision Queue. A Precision Queue Step must also be associated with at least one Precision Attribute.

Creating Precision Queue Steps

When a Precision Queue is first created, at least one Precision Queue Step and the associated memberships must be specified and created at the same time (see section 4.5.17 "Precision Queue Resource Item").

When you create a Precision Queue Step independently of a Precision Queue, you must also:

- specify an existing parent Precision Queue for the Precision Queue Step (the Precision Queue Step Precision Queue Member item that links the queue and the step will be created automatically when the step is created)
- create one or more Precision Queue Step Precision Attribute Member items that link the step that you are creating with the required attributes.

Depending on the position in the queue where you have added the step, you may also need to adjust the properties of the existing steps to maintain the correct numbering sequence and contents.

Deleting Precision Queue Steps

When you delete a Precision Queue Step independently of a Precision Queue, the following related items are deleted automatically:

- the member item that links the step with the queue
- any member items that link attributes with the step that you are about to delete.

Depending on the position in the queue from where you have deleted the step, you may also need to adjust the properties of the remaining steps to maintain the correct numbering sequence.

Note
You cannot delete the last Precision Queue Step in a Precision Queue unless you are also deleting the Precision Queue itself.

Updating Precision Queue Steps

You can use the `Update()` method to update a Precision Queue Step, but note that you cannot change the value specified in the `PrecisionQueueUrn` field. If you want to move a step to a different queue, you must delete the original step and recreate it in the new queue.

Adding Precision Attributes to Precision Queue Steps

To add a Precision Attribute to a Precision Queue Step, create a Precision Queue Step Precision Attribute Member item that links the step to the required attribute. A step can have up to 10 associated attributes.

Deleting Attributes from Precision Queue Steps

To delete a Precision Attribute from a Precision Queue Step, delete the Precision Queue Step Precision Attribute Member item that links the step to the attribute.

Note
You cannot delete the last Precision Queue Step Precision Attribute Member in a Precision Queue Step unless you delete the step itself.

4.5.16.2 Fields

The following fields can be read or set for Precision Queue Step resources:

Table 4.60 Precision Queue Step Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
StepOrder	Int (n-v pair)	The order of rows for a Precision Queue Step. This value must start at 1 (zero is invalid) and increment by 1 for each subsequent step.	Yes	Yes
PrecisionQueueUrn	String (n-v pair)	The id of the parent Precision Queue.	Yes	Yes
WaitTime	Int (n-v pair)	A Wait time to apply before proceeding to the next step (in seconds). The value must be 0 or greater for all steps, excluding the last step. The value for the last step defaults to -1. With a value of -1, the system waits until an agent is available to take the call	No	Yes
ConsiderIf	String (n-v pair)	Consider If expression which must be met in order to execute a particular step. The length of this field is 255 characters. Objects used in the expression are case-sensitive. This field must be null for the last step in a Precision Queue.	No	Yes

4.5.16.3 Associated Name Fields for Precision Queue Steps

As well as the fields above, the following associated name fields can also be used to sort Precision Queue Step resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.61 Associated Name Fields for Sorting Precision Queue Steps

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId

4.5.16.4 Model

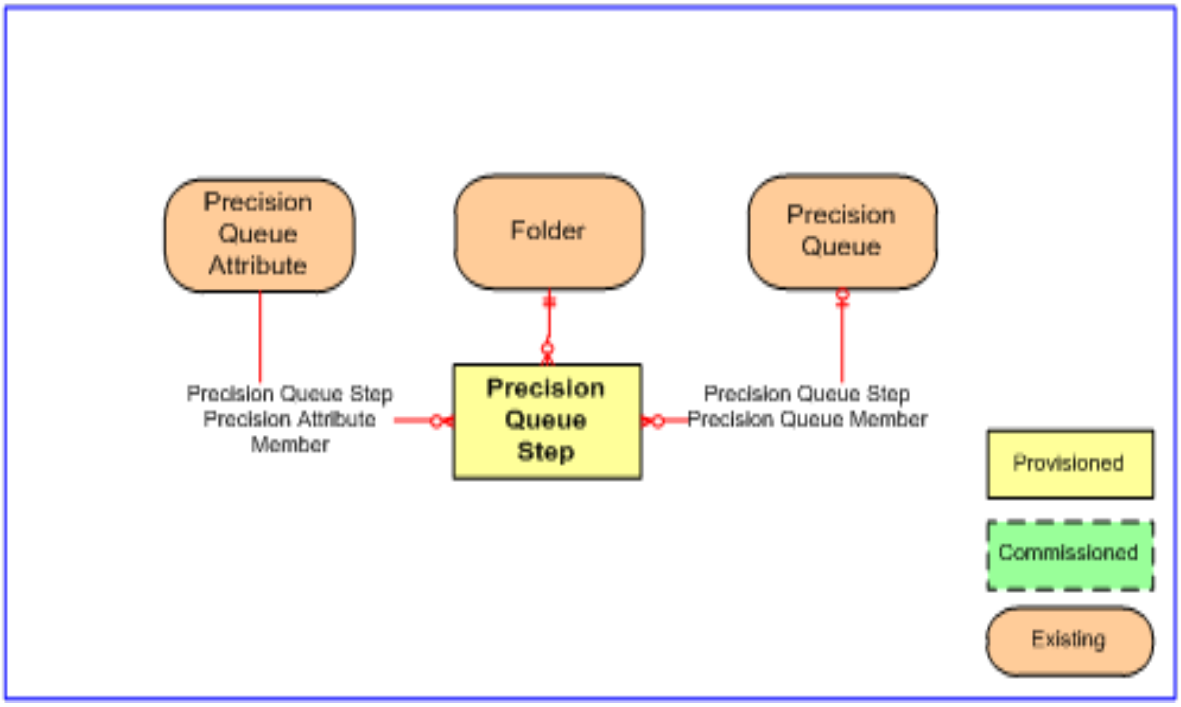


Figure 4.22 Precision Queue Step Relationships

4.5.16.5 REST Protocol

Precision Queue Step Resource REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL]/precision-queue-steps/<id>,<id>/audits?FromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/precision-queue-steps`

Precision Queue Step Resource REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL]/precision-queue-steps/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/precision-queue-step
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/precision-queue-steps/<id>,<id>
Save	Sync.	POST	200 OK	Base URL]/members/precision-queue-steps
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_PRECISION_QUEUE_STEP&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/precision-attributes/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/precision-queue-steps			

4.5.17 Precision Queue Resource Item

4.5.17.1 Description

The `PrecisionQueue` resource defines a queue used for precision routing.

A Precision Queue cannot exist on its own: it is always associated with at least one Precision Queue Step. When you create a queue, you also have to specify at least one step, and a membership between the step and at least one attribute. When the queue is created, the system also creates the specified step or steps and the associated membership items. When you delete a queue, you must also identify and specify the associated steps. The system will also delete these steps and the associated memberships.

Creating Precision Queues

When you create a Precision Queue, you must specify a composite object that includes:

- the Precision Queue,
- at least one Precision Queue Step,

- for each Precision Queue Step that you specify, at least one Precision Queue Step Precision Attribute Member that links the step to an existing attribute.

Optionally, you can specify more than one step and one or more attribute members for each step.

The composite object is passed to the `Create()` method as an ordered array.

Table 4.62 Precision Queue Resource Objects Required by Create()

Resource Type	Resource Id	Description
`PrecisionQueue`	`-1`	The Precision Queue to be created.
`PrecisionQueueStep`	`-1`	The first Precision Queue step to be created and associated with this Precision Queue.
`PrecisionQueueStep PrecisionAttribute Member`	`-3`	The membership that links this Precision Queue Step to the first Precision Attribute in the step. The resource ID of -3 indicates to the server that this should be created at the same as the parent queue step.
`(PrecisionQueueStep PrecisionAttribute Member)`	`-3`	(Optional) Up to nine additional member items linking the step to any remaining attributes in the step.
(Additional Precision Queue Steps and Precision Queue Step Precision Attribute Members)	Various	(Optional) Up to nine additional steps, each followed by between one and ten member items linking the step to the attributes in the step.

When the Precision Queue resource is created, the following items are also created:

- the specified Precision Queue Steps
- for each step, the member that links the step to the queue
- for each step, the specified members or members linking the step to the attributes.

The `Create()` method returns a resource array containing the Precision Queue and the associated child resources with their attributes. The array elements are presented in the same order as they were specified in the call to the `Create()` method. Note that Precision Queue Step Precision Queue Member items are created as required, but are not returned in the resource array.

Deleting Precision Queues

When you delete a Precision Queue you must delete a composite object that includes all the Precision Queue Steps as well as the Precision Queue itself.

The composite object is passed to the `Delete()` method as an ordered array. The `Delete()` method deletes the specified steps and the queue resource. All associated Precision Queue Step Precision Attribute Members and Precision Queue Step Precision Queue Members are also deleted. The following table shows the resource objects required by `Delete()`.

Table 4.63 Precision Queue Resource Objects Required by Delete()

Resource Type	Resource Id	Description
`PrecisionQueueStep`	`<id>`	The Precision Queue Step(s) to be deleted.
`PrecisionQueue`	`<id>`	The Precision Queue to be deleted.

Deleting a Precision Queue: Example

We recommend using the following steps to delete a Precision Queue resource.

1. Select the Precision Queue to be deleted and find its resource key.
2. Find the Precision Queue Steps associated with the Precision Queue using the search query:
   ```
   childof:"Precision Queue",<resourceId> type:"Precision Queue Step" -status:D latest:1
   ```
3. Create a resource array containing the resource keys of the returned Precision Queue Steps.
4. Add the Precision Queue resource key to the end of the resource array.
5. Call the `Delete()` method, passing the resource array.

Adding Precision Queue Steps to Precision Queues

To add a Precision Queue Step to a Precision Queue, create a new Precision Queue Step and specify the id of the Precision Queue in the `PrecisionQueueUrn` field. The member item that links the Precision Queue Step to the required Precision Queue will be created automatically when the Precision Queue Step is created. A Precision Queue can have up to 10 associated Precision Queue Steps.

Deleting Precision Queue Steps from Precision Queues

To delete a Precision Queue Step from a Precision Queue, specify the id of the required Precision Queue Step.

The member item that links the step to the queue will be deleted automatically. All member items linking attributes to the step will also be deleted automatically.

Note

You cannot delete the last Precision Queue Step in a Precision Queue Step unless you delete the Precision Queue itself.

4.5.17.2 Fields

The following fields can be read or set for Precision Queue resources.

Table 4.64 Precision Queue Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionItem`				
`AgentOrdering`	`Int` (n-v pair)	Determines the order of agents in a precision queue sub-queue using the following values: 1 = LAA (agent availability time), 2 = Most skilled agent, 3 = Least skilled agent	Yes	Yes
`CallOrdering`	`Int` (n-v pair)	Determines the order of calls in this precision queue using the following value: 1 = Priority, then time in queue	No	Yes
`ServiceLevelThreshold`	`Int` (n-v pair)	The service level threshold in seconds for this precision queue	Yes	Yes
`ServiceLevelType`	`Short` (n-v pair)	Determines how to calculate the service level for the precision queue using the following values: 1 = ignore abandoned calls, 2 = abandoned call has negative impact, 3 = abandoned call has positive impact	Yes	Yes
`ForceExpandingQueue`	`Bool` (n-v pair)	For future use	No	Yes
`DepartmentBizUrn`	`Int` (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

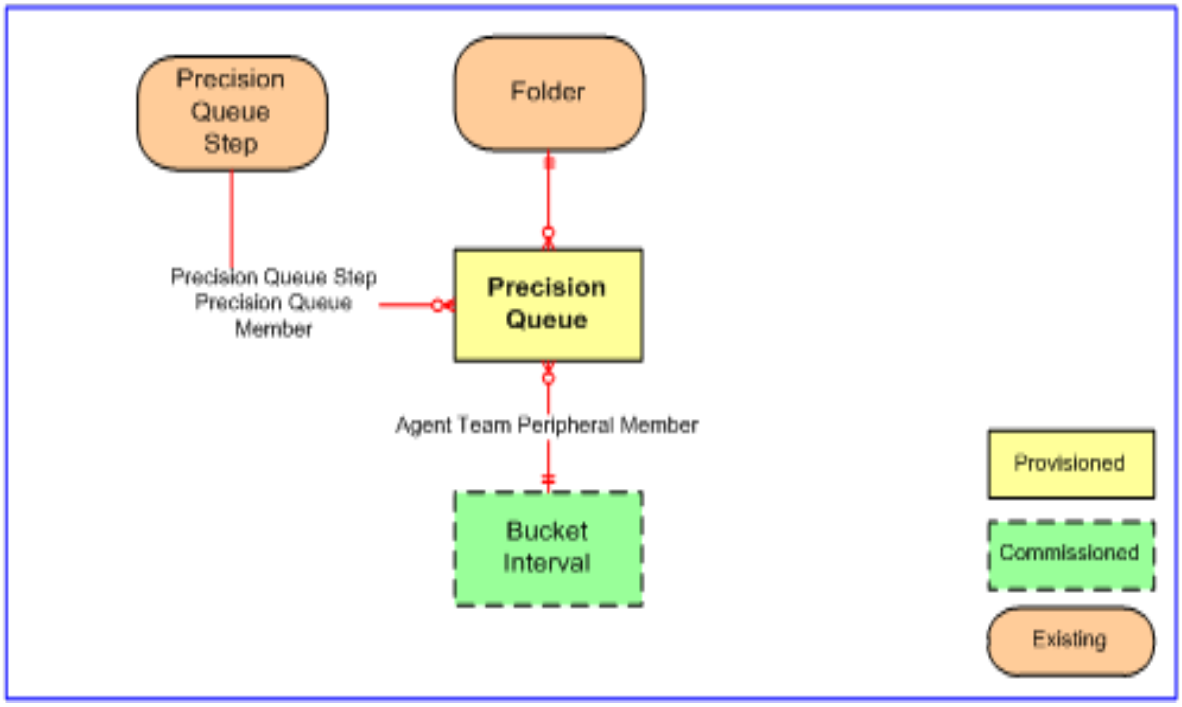
4.5.17.3 Associated Name Fields for Precision Queues

As well as the fields above, the following associated name fields can also be used to sort Precision Queue resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.65 Associated Name Fields for Sorting Precision Queues

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	PrecisionQueue.DepartmentBizUrn

4.5.17.4 Model



4.5.17.5 REST Protocol

Precision Queue Resource REST API Summary	
Base URL	https://<server>:8085/resourcemanagement/rest/resources
IO Format	XML or JSON

Precision Queue Resource REST API Summary				
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL]/precision-queues/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/precision-queues
Delete	Async.	DELETE	202 Accepted	[Base URL]/precision-queues/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/precision-queue
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/precision-queues/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/precision-queues
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_PRECISION_QUEUEmax%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/precision-queues/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/precision-queues			

4.5.18 Route Resource Item

4.5.18.1 Description

The `Route` resource represents any possible destination for a call.

4.5.18.2 Fields

The following fields can be read or set for Route resources.

Table 4.66 Fields in Route Resource Item

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionItem`				
`SkillGroupUrn`	`String` (n-v pair)	The parent skill group for this route.		No

4.5.18.3 Associated Name Fields for Routes

As well as the fields above, the following associated name fields can also be used to sort Route resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.67 Associated Name Fields for Sorting Routes

Associated Name Field	Related Element
`ParentCreatedByLoginName`	`Entry.CreatedById`
`ParentFolderPath`	`Item.FolderId`
`ParentOwnerName`	`DimensionItem.OwnerId`

4.5.18.4 Model

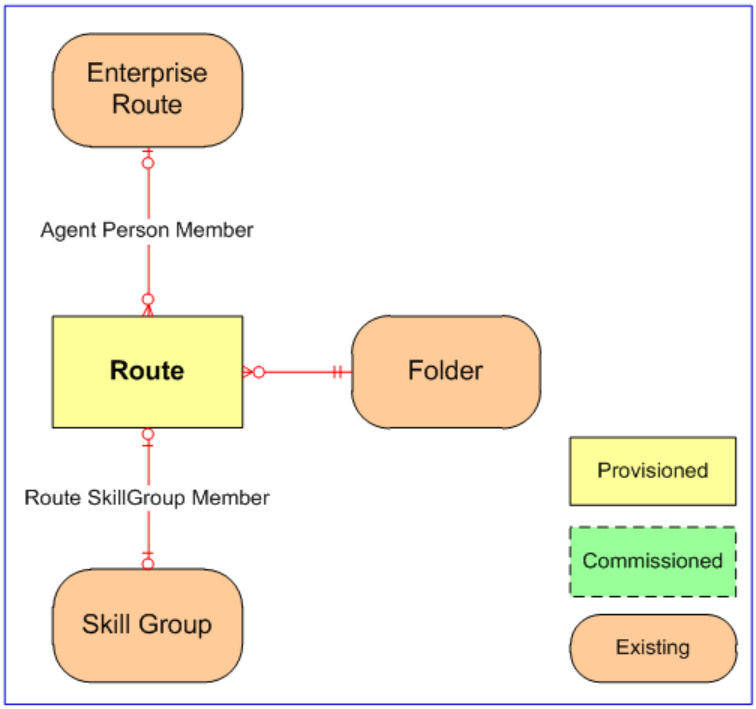


Figure 4.23 Route Relationships

4.5.18.5 REST Protocol

Route Resource REST API Summary				
Base URL	https://<server>:8085/resourcemanagement/rest/resources			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL] /routes/<id>,<id>/audits? fromDate=<fromdate> &toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/routes
Delete	Async.	DELETE	202 Accepted	[Base URL]/routes/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/route

Route Resource REST API Summary				
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/routes/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/routes
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_ROUTE&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/routes/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/routes			

4.5.19 Route Partition Resource Item

4.5.19.1 Description

The RoutePartition is a collection of route patterns that facilitate call routing by dividing the route plan into logical subsets that are based on organization, location, and call type.

4.5.19.2 Fields

The following fields can be read or set for Route Partition resources:

Table 4.68 Route Partition Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
No further fields.				

4.5.19.3 Associated Name Fields for Route Partitions

As well as the fields above, the following associated name fields can also be used to sort Route Partition resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.69 Associated Name Fields for Sorting Route Partitions

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId

4.5.19.4 Model

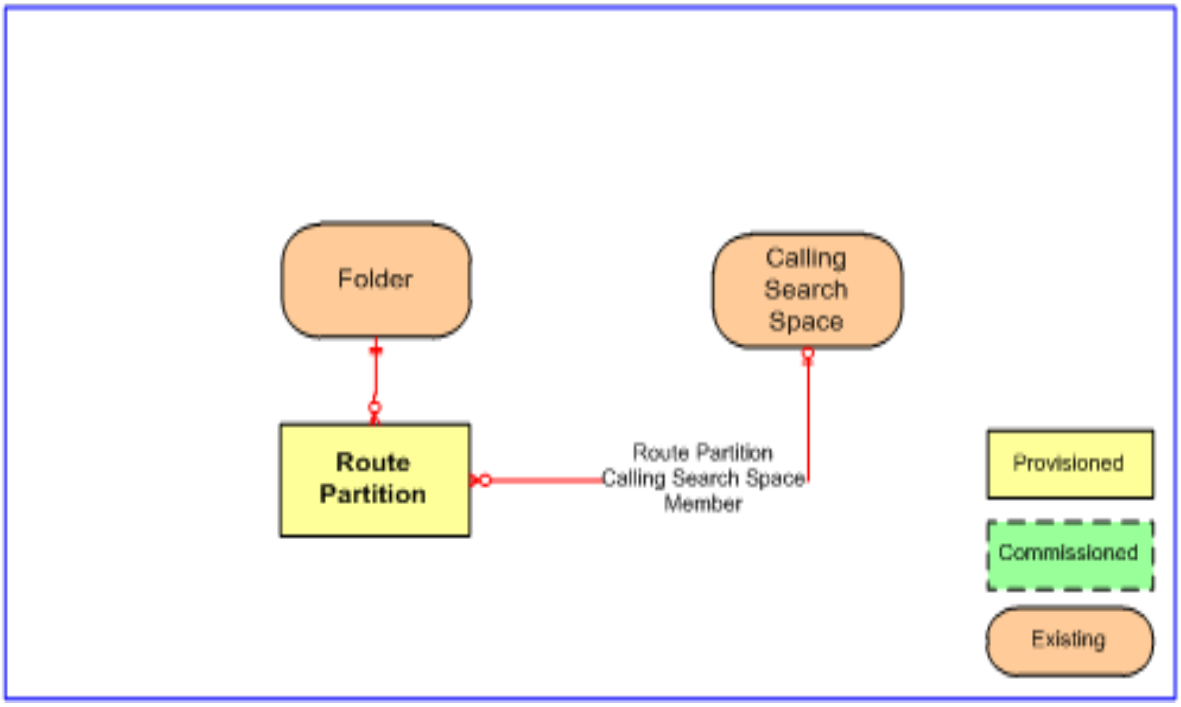


Figure 4.24 Route Partition Relationships

4.5.19.5 REST Protocol

Route Partition Resource REST API Summary				
Base URL	https://<server>:8085/resourcemanagement/rest/resources			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL]/route-partitions/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/route-partitions
Delete	Async.	DELETE	202 Accepted	[Base URL]/route-partitions/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/route-partition
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/route-partitions/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/route-partitions
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_ROUTE_PARTITION&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/route-partitions/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/route-partitions			

4.5.20 Service Resource Item

4.5.20.1 Description

The Service resource represents a service on a peripheral.

4.5.20.2 Fields

The following fields can be read or set for Service resources:

Table 4.70 Service Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
PeripheralUrn	String (10) (n-v pair)	The peripheral or equipment identifier on which the service will be located. If -1 is supplied then Unified CCDM will pick the Peripheral automatically. Note: cannot be updated after the Create.	Yes	No
MediaRoutingDomainUrn	String (10) (n-v pair)	The media routing domain that services this service.	Yes	No
PeripheralNumber	Int (n-v pair)	This is the number for the service as known to the peripheral it is associated with.	No	Yes
PeripheralName	String (n-v pair)	This is the name of the service as known to the peripheral it is associated with.	Yes	Yes
PeripheralServiceLevelType	Short (n-v pair)	This is the type of service level used for the service on the peripheral.	Yes	Yes
ServiceLevelType	Short (n-v pair)	This is the type that defines how service levels are calculated for the service. This take the values 0: Default; 1: Ignore abandoned calls; 2: Abandoned calls have negative impact; 3: Abandoned calls have positive impact.	Yes	Yes
ServiceLevelThreshold	Int (n-v pair)	This is the threshold in seconds for the service.	Yes	Yes
Extension	String	This is the extension number for the skill group.	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
UserDeletable	Bool	Indicates if the item can be deleted (used for parent/child configurations)	No	Yes
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.20.3 Associated Name Fields for Services

As well as the fields above, the following associated name fields can also be used to sort Service resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.71 Associated Name Fields for Sorting Services

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	Service.DepartmentBizUrn
ParentPeripheralDisplayName	Service.PeripheralUrn

4.5.20.4 Model

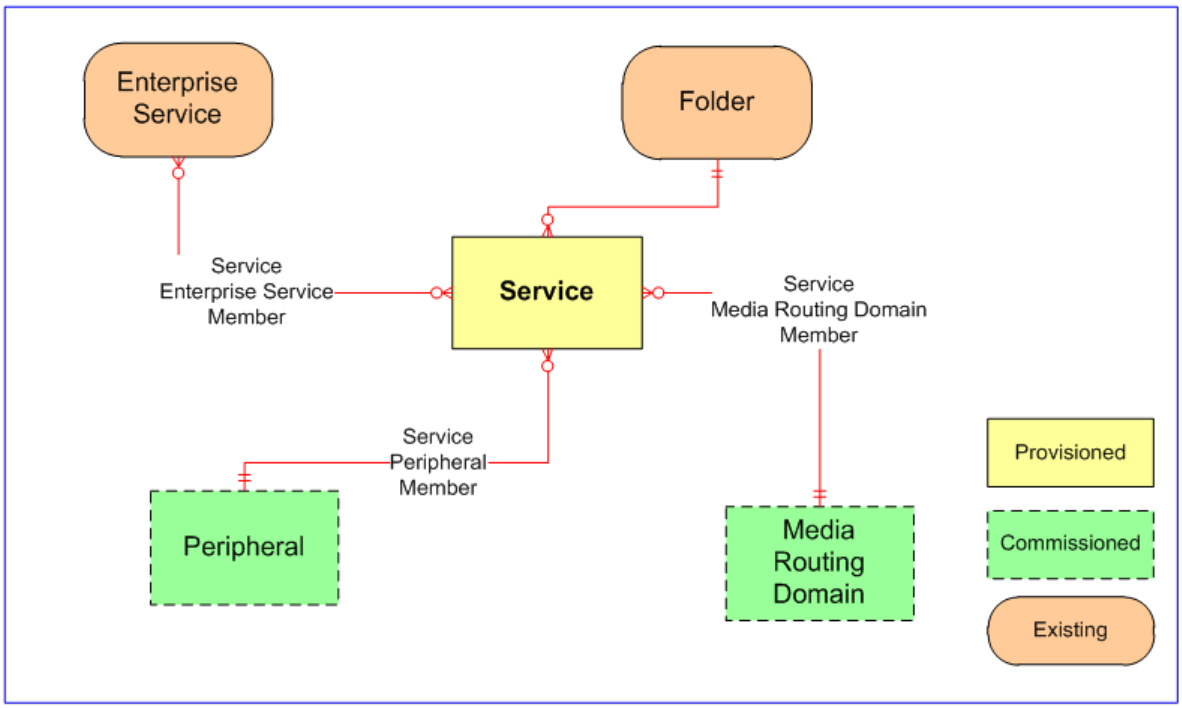


Figure 4.25 Service Relationships

4.5.20.5 REST Protocol

Service Resource REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL] /services/<id>,<id>/audits? fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/services`

Service Resource REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL]/services/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/services
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/services/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/services
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_SERVICE&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/services/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/services			

4.5.21 Skill Group Resource Item

4.5.21.1 Description

The SkillGroup resource represents a queue point in the contact routing solution to which agents are added for the purpose of dealing with the work items, for example, inbound voice calls, emails tasks etc.

4.5.21.2 Fields

The following fields may be read or set on skill group resource objects.

Table 4.72 Fields in Skill Group Resource Item

Element Name	Data Type	Description	Required	Pkey
Inherits From: DimensionItem				
PeripheralUrn	String (10) (n-v pair)	The peripheral or equipment identifier on which the skill group will be located. If -1 -s supplied then Unified CCDM will pick the Peripheral automatically. Note: cannot be updated after the Create.	Yes	No

Element Name	Data Type	Description	Required	Pkey
MediaRoutingDomainUrn	String (10) (n-v pair)	The media routing domain to be used by this skillgroup. Note: cannot be modified using the Update call.	Yes	No
PeripheralNumber	String (50) (n-v pair)	This is the number for the skillgroup as known to the peripheral it is associated with.	No	Yes
PeripheralName	String (50) (n-v pair)	This is the name for the skillgroup as known to the peripheral it is associated with	No	Yes
AvailableHoldoff Delay	Short (n-v pair)	This is the number of seconds before an agent becomes available after a call is terminated. Defaults to 0.	Yes	Yes
Priority	Short (n-v pair)	This is the routing priority for the skillgroup. This can take the values 1: Primary; 2: Secondary; 3: Tertiary	Yes	Yes
Extension	String (50) (n-v pair)	This is the extension number for the service	No	Yes
IPTA	Char (n-v pair)	This is a character flag (Y or N) indicating whether for this skill group the routing platform picks the agent. Defaults to N. Note: Only supported on Unified CCE.	No	Yes
ServiceLevelThreshold	Int (n-v pair)	This is the service level threshold for the skill group. Defaults to -1 meaning that if not set the value defaults to that set by the peripheral.	No	Yes
ServiceLevelType	Short (n-v pair)	This is the type that defines how service levels are calculated for the skillgroup. Defaults to 0. This takes the values: 0: Default; 1: Ignore abandoned calls; 2: Abandoned calls have negative impact; 3: Abandoned calls have positive impact.	No	Yes
DefaultEntry	String (50) (n-v pair)	This is the default entry for the skillgroup	No	Yes

Element Name	Data Type	Description	Required	Pkey
SubSkillGroupMask	String (64) (n-v pair)	A series of characters (Y and N) indicating which sub-skillgroups to create for the skillgroup.	No	Yes
UserDeletable	Bool (n-v pair)	Indicates if the item can be deleted by end users as opposed to the Unified CCDM background processed. Note: used only for Unified CCE parent/child configurations only)	No	Yes
IsPRSkill	Boolean (n-v pair)	(Read only). Indicates whether this skill group is a precision routing skill group.	No	Yes
DepartmentBizUrn	Int (n-v pair)	Foreign key to the department that this resource is associated with, or -1 if there is no associated department.	No	No

4.5.21.3 Associated Name Fields for Skill Groups

As well as the fields above, the following associated name fields can also be used to sort Skill Group resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.73 Associated Name Fields for Sorting Skill Groups

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId
ParentDepartmentDisplayName	SkillGroup.DepartmentBizUrn
ParentPeripheralDisplayName	SkillGroup.PeripheralUrn

4.5.21.4 Model

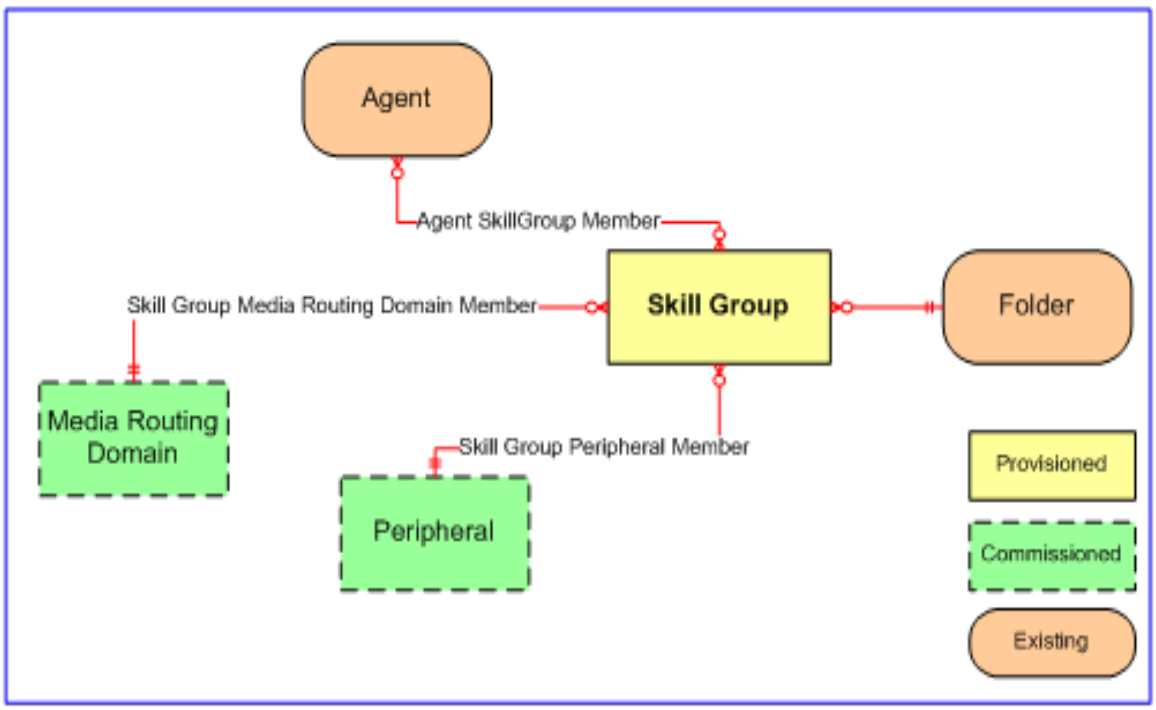


Figure 4.26 Skill Group Relationships

4.5.21.5 REST Protocol

Skill Group Resource REST API Summary				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	`GET`	`200 OK`	`[Base URL] /tenants/<id>,<id>/audits? fromDate=<fromdate>&toDate=<toDate>…`
Create	Async.	`POST`	`202 Accepted`	`[Base URL]/skillgroups`

Skill Group Resource REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL] /skillgroups/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/tenant
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL] /skillgroups/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/skillgroups
Search	Sync.	GET	200 OK	[Base URL]?queryString= type%3aIT_SKILLGROUP&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL] /skillgroups/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/skillgroups			

4.5.22 Tenant Resource Item

4.5.22.1 Description

The Tenant resource represents a company or organization. A tenant may exist on dedicated equipment, for example, a dedicated physical or virtualized stack, or may share a portion of an equipment stack. The key definition of a tenant is that is must have telephony/CTI isolation from other tenants on the same equipment stack. For example, in a Cisco UCCE environment, the use of Calling Search Spaces and Route Partitions enforce that Tenant A extensions can't be reached by Tenant B.

4.5.22.2 Fields

Fields that may be set or read on a Tenant resource item are:

Table 4.74 Fields in Tenant Resource Item

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
No further fields.				

4.5.22.3 Associated Name Fields for Tenants

As well as the fields above, the following associated name fields can also be used to sort Tenant resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.75 Associated Name Fields for Sorting Tenants

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId

4.5.22.4 Model

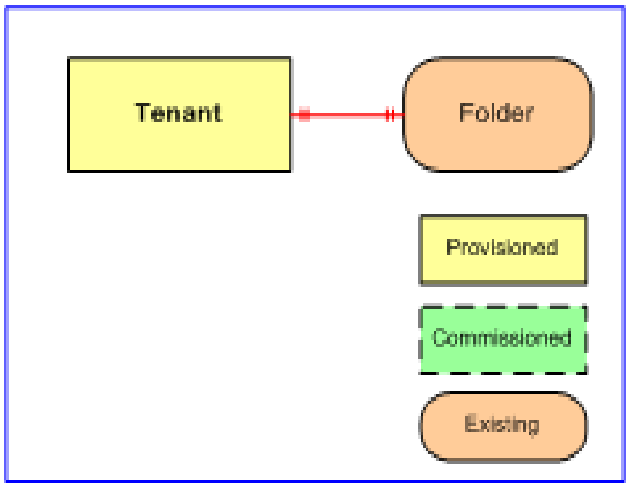


Figure 4.27 Tenant Relationships

4.5.22.5 REST Protocol

Tenant Resource REST API Summary	
Base URL	https://<server>:8085/resourcemanagement/rest/resources
IO Format	XML or JSON

Tenant Resource REST API Summary				
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL] /tenants/<id>,<id>/audits? fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/tenants
Delete	Async.	DELETE	202 Accepted	[Base URL]/tenants/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/tenant
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/tenants/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/tenants
Search	Sync.	GET	200 OK	[Base URL]?queryString= type%3aIT_TENANT&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/tenants/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/tenants			

4.5.23 User Variable Partition Resource Item

4.5.23.1 Description

The UserVariable resource is a user defined variable that can be used in the call routing process.

4.5.23.2 Fields

The following fields can be read or set for User Variable resources:

Table 4.76 User Variable Resource Item Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionItem				
ObjectType	Guid (n-v pair)	This is a number that indicates the object associated with the user variable. These objects can be skill groups, services etc. The values correspond to the values in OBJECT_TYPE_CODE in TB_DIM_OBJECT_TYPE.	Yes	Yes
DataType	Short (n-v pair)	This a number indicating the data type for the user variable. This can take values 0: Long; 1: Float; 2: Char; 3: Date.	Yes	Yes
Persistent	Char (n-v pair)	This is a character flag indicating whether to preserve the value of the user variable between script invocations. Value Y indicates yes.	Yes	Yes

4.5.23.3 Associated Name Fields for User Variables

As well as the fields above, the following associated name fields can also be used to sort User Variable resource items returned by the search API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.77 Associated Name Fields for Sorting User Variables

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId

4.5.23.4 Model

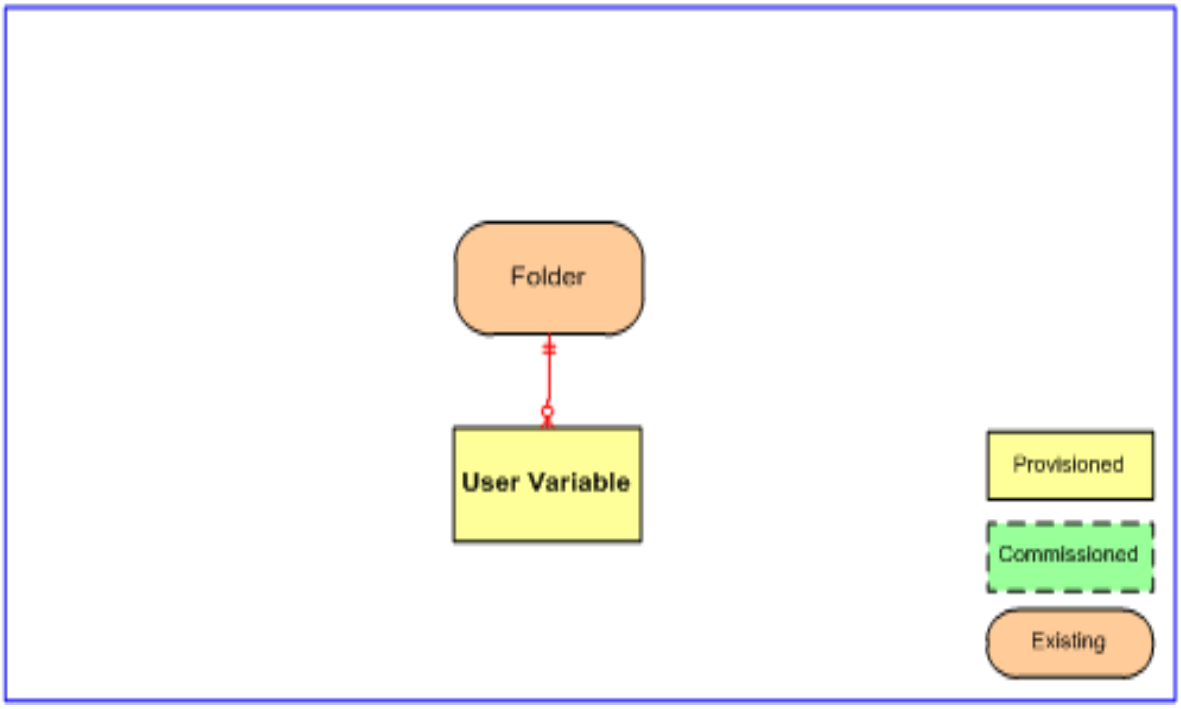


Figure 4.28 User Variable Relationships

4.5.23.5 REST Protocol

User Variable Resource REST API Summary				
Base URL	https://<server>:8085/resourcemanagement/rest/resources			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**
Audit	Sync.	GET	200 OK	[Base URL]/user-variables /<id>,<id>/audits? fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/user-variables

User Variable Resource REST API Summary				
Delete	Async.	DELETE	202 Accepted	[Base URL]/user-variables/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/user-variable
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/user-variables/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/user-variables
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_USER_VARIABLE&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/user-variables/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Example	POST https://Web01:8085/resourcemanagement/rest/resources/user-variables			

4.6 Non-Provisionable Remote Resource Types

4.6.1 About Non-Provisionable Remote Resources

The non-provisionable remote resource types are listed in section 3.1.3 "Non Provisionable Remote Resource Types". These resource types are supported for searching and retrieving only.

4.6.2 Associated Name Fields for Non-Provisionable Remote Resources

The following associated name fields can also be used to sort non-provisionable remote resource items returned by the `search` API. The table shows the associated name field and the related element that is used to look up the name.

Table 4.78 Associated Name Fields for Sorting Non-Provisionable Remote Resources

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById
ParentFolderPath	Item.FolderId
ParentOwnerName	DimensionItem.OwnerId

4.7 Resource Management Web Service Member Types

4.7.1 About Resource Management Web Service Member Types

4.7.1.1 Description

Member types define memberships between entities.

4.7.1.2 Member Types and Identifiers

Table 4.79 Member Types and Identifiers

Member Type	Internal Name	REST Parameter
Agent Agent Desktop Member	MT_AGENT_AGENT_DESKTOP_MEMBER	agent-agent-desktop-member
Agent Agent Team Member	MT_AGENT_AGENT_TEAM_MEMBER	agent-agent-team-member
Agent Desktop Dialed Number Member	MT_AGENT_DESKTOP_DIALED_NUMBER_MEMBER	agent-desktop-dialed-number-member
Agent Peripheral Member	MT_AGENT_PERIPHERAL_MEMBER	agent-peripheral-member
Agent Person Member	MT_AGENT_PERSON_MEMBER	agent-person-member
Agent Precision Attribute Member	MT_AGENT_PRECISION_ATTRIBUTE_MEMBER	agent-precision-attribute-member
Agent Skill Group Member	MT_AGENT_SKILLGROUP_MEMBER	agent-skill-group-member
Agent Team Dialed Number Member	MT_AGENT_TEAM_DIALED_NUMBER_MEMBER	agent-dialed-number-member
Call Type Routing Script Member	MT_CALL_TYPE_ROUTING_SCRIPT_MEMBER	call-type-routing-script-member
Dialed Number Call Type Member	MT_DIALED_NUMBER_CALL_TYPE_MEMBER	dialed-number-call-type-member
Dialed Number Routing Client Member	MT_DIALED_NUMBER_ROUTING_CLIENT_MEMBER	dialed-number-routing-client-member
Dialed Number Media Routing Domain Member	MT_DIALED_NUMBER_MEDIA_ROUTING_DOMAIN_MEMBER	dialed-number-media-routing-domain-member
Group Group Member	MT_GROUP_GROUP_MEMBER	group-group-member

Member Type	Internal Name	REST Parameter
Item Category Member	MT_ITEM_CATEGORY_MEMBER	item-category-member
Label Dialed Number Member	MT_LABEL_DIALED_NUMBER_MEMBER	label-dialed-number-member
Label Routing Client Member	MT_LABEL_ROUTING_CLIENT_MEMBER	label-routing-client-member
Network Vru Script Network Vru Member	MT_NETWORK_VRU_SCRIPT_NETWORK_VRU_MEMBER	network-vru-script-network-vru-member
Precision Queue Bucket Interval Member	MT_PRECISION_QUEUE_BUCKET_INTERVAL_MEMBER	precision-queue-bucket-interval-member
Precision Queue Step Precision Attribute Member	MT_PRECISION_QUEUE_STEP_PRECISION_ATTRIBUTE_MEMBER	precision-queue-step-precision-attribute-member
Precision Queue Step Precision Queue Member	MT_PRECISION_QUEUE_STEP_PRECISION_QUEUE_MEMBER	precision-queue-step-precision-queue-member
Query Rule Campaign Member	MT_QUERY_RULE_CAMPAIGN_MEMBER	query-rule-campaign-member-member
Route Partition Calling Search Space Member	MT_ROUTE_PARTITION_CALLING_SEARCH_SPACE_MEMBER	route-partition-calling-search-space-member
Route Skill Group Member	MT_ROUTE_SKILLGROUP_MEMBER	route-skill-group-member
Service Enterprise Service Member	MT_SERVICE_ENTERPRISE_SERVICE_MEMBER	service-enterprise-service-member
Service Media Routing Domain Member	MT_SERVICE_MEDIA_ROUTING_DOMAIN_MEMBER	service-media-routing-domain-member
Service Peripheral Member	MT_SERVICE_PERIPHERAL_MEMBER	service-peripheral-member
Skill Group Campaign Member	MT_SKILLGROUP_CAMPAIGN_MEMBER	skill-group-campaign-member
Skill Group Enterprise Skill Group Member	MT_SKILLGROUP_ENTERPRISE_SKILLGROUP_MEMBER	skill-group-enterprise-skill-group-member

Member Type	Internal Name	REST Parameter
Skill Group Media Routing Domain Member	MT_SKILLGROUP_MEDIA_ROUTING_DOMAIN_MEMBER	skill-group-media-routing-domain-member
Skill Group Peripheral Member	MT_SKILLGROUP_PERIPHERAL_MEMBER	skill-group-peripheral-member
Skill Group Service Member	MT_SKILLGROUP_SERVICE_MEMBER	skill-group-service-member
User Group Member	MT_USER_GROUP_MEMBER	user-group-member

4.7.1.3 Member Operations and Relationships

This table lists the supported member types, the operations supported by each member type, and the parent and child items which define each member type.

Table 4.80 Member Operations and Relationships

Resource Member Type	Create	Edit	Delete	Read	Parent	Child
Agent Agent Desktop Member	False	False	False	True	Agent Desktop	Agent
Agent Agent Team Member	True	True	True	True	Agent Team	Agent
Agent Desktop Dialed Number Member	True	True	True	True	Dialed Number	Agent Desktop
Agent Peripheral Member	False	False	False	True	Peripheral	Agent
Agent Person Member	False	True	False	True	Person	Agent
Agent Precision Attribute Member	True	True	True	True	Precision Attribute	Agent
Agent Team Dialed Number Member	True	True	True	True	Dialed Number	Agent Team
Agent Skill Group Member	True	False	True	True	Skill Group	Agent
Call Type Routing Script Member	True	True	True	True	Routing Script	Call Type
Dialed Number Call Type Member	True	True	True	True	Call Type	Dialed Number
Dialed Number Routing Client Member	False	False	False	True	Routing Client	Dialed Number
Dialed Number Media Routing Domain Member	False	False	False	True	Media Routing Domain	Dialed Number

Resource Member Type	Create	Edit	Delete	Read	Parent	Child
Group Group Member	True	True	True	True	Group	Group
Item Category Member	True	True	True	True	Category	Any Dimension Item
Label Dialed Number Member	True	True	True	True	Dialed Number	Label
Label Routing Client Member	False	False	False	True	Routing Client	Label
Network Vru Script Network Vru Member	False	True	False	True	Network Vru	Network Vru Script
Precision Queue Bucket Interval Member	True	True	True	True	Bucket Interval	Precision Queue
Precision Queue Step Precision Attribute Member	True	True	True	True	Precision Attribute	Precision Queue Step
Precision Queue Step Precision Queue Member	False	False	False	True	Precision Queue	Precision Queue Step
Route Partition Calling Search Space Member	True	True	True	True	Calling Search Space	Route Partition
Route Skill Group Member	True	True	True	True	Skill Group	Route
Service Enterprise Service Member	False	False	False	True	Enterprise Service	Service
Service Media Routing Domain Member	False	False	False	True	Media Routing Domain	Service
Service Peripheral Member	False	False	False	True	Peripheral	Service
Skill Group Enterprise Skill Group Member	True	True	True	True	Enterprise Skill Group	Skill Group
Skill Group Media Routing Domain Member	False	False	False	True	Media Routing Domain	Skill Group
Skill Group Peripheral Member	False	False	False	True	Peripheral	Skill Group
Skill Group Service Member	True	True	True	True	Service	Skill Group
User Group Member	True	True	True	True	Group	User

4.7.1.4 Member Pkey Types

Table 4.81 Pkey Member Types

Pkey Member Type	Create	Edit	Delete	Read	Parent	Child
Agent Agent Team Member Pkey	True	True	True	True	Agent Team	Agent
Agent Peripheral Member Pkey	True	True	True	True	Peripheral	Agent
Agent Precision Attribute Member Pkey	True	True	True	True	Precision Attribute	Agent
Agent Skill Group Member Pkey	True	True	True	True	Skill Group	Agent
Call Type Routing Script Member Pkey	True	True	True	True	Routing Script	Call Type
Dialed Number Call Type Member Pkey	True	True	True	True	Call Type	Dialed Number
Precision Queue Step Precision Attribute Member Pkey	True	True	True	True	Precision Attribute	Precision Queue
Query Rule Campaign Member Pkey	True	True	True	True	Campaign	Query Rule
Route Partition Calling Search Space Member Pkey	True	True	True	True	Calling Search Space	Route Partition
Skill Group Campaign Member Pkey	True	True	True	True	Campaign	Skill Group
Skill Group Service Member Pkey	True	True	True	True	Service	Skill Group

4.7.1.5 REST Protocol

Table 4.82 describes the REST APIs for all member types.

To derive the REST URL for a specific member type, replace `<rest resource>` and `<internal name>` with the corresponding REST parameter and internal name for the member type from Table 4.79 .

See Table 4.80 for the operations that are valid for each member type.

Table 4.82 REST API Specification

REST API Specification				
Base URL	`https://<server>:8085/resourcemanagement/rest/resources`			
IO Format	XML or JSON			
Operation	**Mode**	**Method**	**Status**	**URL**

REST API Specification				
Audit	Sync.	GET	200 OK	[Base URL]/<rest resource>s/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	200 OK	[Base URL]/<rest resource>s
Delete	Async.	DELETE	202 Accepted	[Base URL]/<rest resource>s/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/<rest resource>
Move	n/a	n/a	n/a	n/a
Retrieve	Sync.	GET	200 OK	[Base URL]/<rest resource>s/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/<rest resource>s
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3a<internal name>&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/<rest resource>s/<id>,<id>
Exceptions	See section 4.2.3 "Errors"			
Examples	POST https://Web01:8085/resourcemanagement/rest/resources/gadget-provider-user-mappings GET https://Web01:8085/resourcemanagement/rest/resources/queryString=type%3aMT_AGENT_AGENT_TEAM_MEMBERmax%3a10			

4.7.1.6 Creating New Members

To create a new member, pass a new resource object to the `Create()` method with the type set to the appropriate `ResourceMemberType` and with the parent and child ids set accordingly.

Note

Precision Queue Step Precision Queue Members cannot be created explicitly using `Create()`. They are always created automatically when the corresponding Precision Queue Step is created. This is because each Precision Queue Step must always be associated with a Precision Queue: a Precision Queue Step cannot exist separately from a Precision Queue.

4.7.1.7 Associated Name Fields for Members

As well as the fields listed below for each member type, the following associated name field can also be used to sort members returned by the `search` API. The table shows the associated name field and the related member element that is used to look up the name.

Table 4.83 Associated Name Field for Sorting Members

Associated Name Field	Related Element
`ParentCreatedByLoginName`	`Entry.CreatedById`

4.7.2 Agent Agent Desktop Member

4.7.2.1 Description

The Agent Agent Desktop Member defines a membership between an Agent and an Agent Desktop. It is a many-one relationship and each Agent can only be associated with one Agent Desktop.

4.7.2.2 Fields

The `AgentAgentDesktopMember` class has the following fields:

Table 4.84 Agent Agent Desktop Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.3 Agent Agent Team Member

4.7.3.1 Description

The Agent Agent Team Member defines a membership between an Agent and an Agent Team. It is a many-many relationship.

4.7.3.2 Fields

The `AgentAgentTeamMember` class has the following fields:

Table 4.85 Agent Agent Team Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
Supervisor	Bool (n-v pair)	Indicates whether the agent is a supervisor of the agent team.	No	Yes
PrimarySupervisor	Bool (n-v pair)	Indicates whether the agent is the primary supervisor of the agent team.	No	Yes
PhysicalMember	Bool (n-v pair)	Indicates whether the agent is a physical member of the agent team (an agent may be associated with an agent team as a supervisor, but may not be a physical team member).	No	Yes

Note
At least one of Supervisor, PrimarySupervisor and PhysicalMember must be set.

4.7.4 Agent Desktop Dialed Number Member

4.7.4.1 Description

The Agent Desktop Dialed Number Member defines a membership between an Agent Desktop and a Dialed Number. It is a many-one relationship and each Agent Desktop can only be associated with one Dialed Number.

4.7.4.2 Fields

The AgentDesktopDialedNumberMember class has the following fields:

Table 4.86 Agent Desktop Dialed Number Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
No further fields.				

4.7.5 Agent Peripheral Member

4.7.5.1 Description

The Agent Peripheral defines a membership between an Agent and a Peripheral. It is a many-many relationship.

4.7.5.2 Fields

The `AgentPeripheralMember` class has the following fields:

Table 4.87 Agent Peripheral Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.6 Agent Person Member

4.7.6.1 Description

The Agent Person Member defines a membership between an Agent and a Person. It is a many-one relationship, and each Agent can only be associated with one Person.

4.7.6.2 Fields

The `AgentPersonMember` class has the following fields:

Table 4.88 Department Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.7 Agent Precision Attribute Member

4.7.7.1 Description

The Agent Precision Attribute Member defines a membership between an Agent and a Precision Attribute. It is a many-many relationship.

4.7.7.2 Fields

The `AgentPrecisionAtributeMember` class has the following fields:

Table 4.89 Agent Precision Attribute Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
AttributeValue	String (255) (n-v pair)	The value of the attribute for the agent. Must be convertible to the data type of the attribute. If not supplied, will that the default value for that attribute.	No	Yes
Description	String (255) (n-v pair)	A string containing additional information about the membership.	No	Yes

4.7.8 Agent Team Dialed Number Member

4.7.8.1 Description

The Agent Team Dialed Number Member defines a membership between an Agent Team and a Dialed Number. It is a many-one and each Agent Team can only be associated with one Dialed Number.

4.7.8.2 Fields

The AgentTeamDialedNumberMember class has the following fields:

Table 4.90 Agent Team Dialed Number Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
No further fields.				

4.7.9 Agent Skill Group Member

4.7.9.1 Description

The Agent Skill Group Member defines a membership between an Agent and a Skill Group. It is a many-many relationship.

4.7.9.2 Fields

The AgentSkillgroupMember class has the following fields:

Table 4.91 Agent Skill Group Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
`DefaultMember`	`Bool` (n-v pair)	Indicates if this is the default skill group for the agent.	No	Yes

4.7.10 Call Type Routing Script Member

4.7.10.1 Description

The Call Type Routing Script Member defines a membership between a Call Type and a Routing Script. It is a many-many relationship.

4.7.10.2 Fields

The `CallTypeRoutingMember` class has the following fields:

Table 4.92 Call Type Routing Script Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.11 Dialed Number Call Type Member

4.7.11.1 Description

The Dialed Number Call Type Member defines a relationship between a Dialed number and Call Type. It is a many-many relationship.

4.7.11.2 Fields

The `DialedNumberCallTypeMember` class has the following fields:

Table 4.93 Dialed Number Call Type Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
RegionBizUrn	Int (n-v pair)	The URN as in TB_DIM_REGION.	No	No
Item	Int (n-v pair)	This is a sequence in which rows for a dialed number are tested against the call qualifiers.	Yes	No
CLIWildCard	Int (n-v pair)	A region name or a prefix indicating the leading digits of a telephone number. This can also be a complete telephone number.	No	Yes
CLIWildCardType	Int (n-v pair)	The type of the CLI wild card. One of: • 0 Unknown • 1 NPA (3 digit match) • 2 NPA-NXX (6 digit match) • 3 Match(all digit match) • 4 Region • 5 All (match all CLIs) • 6 Prefix	No	Yes
CEDWildCard	String (60)	A wild card value to be matched with the CED. One of: • _A All • _NR Not required • _NE Not entered • _N None required or entered.	No	Yes
Description	String (255) (n-v pair)	A string containing any additional information about the membership.	No	Yes

4.7.12 Dialed Number Routing Client Member

4.7.12.1 Description

The Dialed Number Routing Client Member defines a relationship between a Dialed Number and a Routing Client. It is a many-one relationship and each Dialed Number can only be associated with one Routing Client.

4.7.12.2 Fields

The `DialedNumberRoutingClientMember` class has the following fields:

Table 4.94 Dialed Number Routing Client Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.13 Dialed Number Media Routing Domain Member

4.7.13.1 Description

The Dialed Number Media Routing Domain Member defines a relationship between a Dialed Number and a Media Routing Domain. It is a many-one relationship and each Dialed Number can only be associated with one Media Routing Domain.

4.7.13.2 Fields

The `DialedNumberMediaRoutingDomainMember` class has the following fields:

Table 4.95 Dialed Number Media Routing Domain Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.14 Group Group Member

4.7.14.1 Description

The Group Group Member defines a membership between a child Group and a parent Group . It is a many-many relationship although a Group cannot have a membership with itself, nor can the same Group be both a parent and child of another Group.

4.7.14.2 Fields

The `GroupGroupMember` class has the following fields:

Table 4.96 Group Group Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: SystemMember				
No further fields.				

4.7.15 Label Dialed Number Member

4.7.15.1 Description

The Label Dialed Number Member defines a relationship between a Label and a Dialed Number. It is a many-many relationship.

4.7.15.2 Fields

The LabelDialedNumberMember class has the following fields:

Table 4.97 Label Dialed Number Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
No further fields.				

4.7.16 Label Routing Client Member

4.7.16.1 Description

The Label Routing Client Member defines a relationship between a Label and a Routing Client. It is a many-one relationship and each Label can only be associated with one Routing Client.

4.7.16.2 Fields

The LabelRoutingClientMember class has the following fields:

Table 4.98 Label Routing Client Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
No further fields.				

4.7.17 Network Vru Script Network Vru Member

4.7.17.1 Description

The Network Vru Script Network Vru Member defines a relationship between an Network VRU Script and a Network VRU. It is a many-one relationship and each Network VRU Script can only be associated with one Network VRU.

4.7.17.2 Fields

The `NetworkVruScriptNetworkVruMember` class has the following fields:

Table 4.99 Network Vru Script Network Vru Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.18 Precision Queue Bucket Interval Member

4.7.18.1 Description

The Precision Queue Bucket Interval Member defines a relationship between a Precision Queue and a Bucket Interval. It is a many-one relationship and each Precision Queue can only be associated with one Bucket Interval.

4.7.18.2 Fields

The `PrecisionQueueBucketInterval` class has the following fields:

Table 4.100 Precision Queue Bucket Interval Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.19 Precision Queue Step Precision Attribute Member

4.7.19.1 Description

The Precision Queue Step Precision Attribute Member defines a relationship between an Precision Queue Step and a Precision Queue. It is a many-many relationship.

4.7.19.2 Fields

The PrecisionQueueStepPrecisionAttributeMember class has the following fields:

Table 4.101 Precision Queue Step Precision Attribute Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
Value1	String (255) (n-v pair)	The value that the attribute is tested against. It must be able to be converted to the AttributeDataType specified for the related Precision Attribute.	Yes	Yes
TermOrder	Int (n-v pair)	The order of the terms in a Precision Queue Step. This value must start at 1 (zero is invalid) for each new Precision Queue Step and increment by 1 for each subsequent term in the Precision Queue Step.	Yes	No
ParenthesesCount	Int (n-v pair)	The number of parenthesis around this term. A positive number indicates open parentheses before the term; a negative number indicates closed parentheses after the term. The total value of ParenthesesCount for all Precision Queue Step Precision Attribute Member entries for a Precision Queue Step must be 0.	Yes	Yes
TermRelation	Int (n-v pair)	Indicates the relationship of this term to the preceding term. One of: 0 None (only valid for the first term in the Precision Queue Step) 1 AND 2 OR	Yes	Yes
AttributeRelation	Int (n-v pair)	Indicates the kind of comparison to be done on the attribute. One of: 1 equals 2 not equal to 3 less than 4 less than or equal to 5 greater than 6 greater than or equal to	Yes	Yes

4.7.20 Precision Queue Step Precision Queue Member

4.7.20.1 Description

The Precision Queue Step Precision Queue defines a relationship between a Precision Queue Step and a Precision Queue. It is a many-one relationship and each Precision Queue Step must be associated with exactly one Precision Queue.

4.7.20.2 Fields

The `PrecisionQueueStepPrecisionQueueMember` class has the following fields:

Table 4.102 Precision Queue Step Precision Queue Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.21 Route Partition Calling Search Space Member

4.7.21.1 Description

The Route Partition Calling Search Space Member defines a relationship between a Route Partition and a Calling Search Space. It is a many-one relationship and each Route Partition can only be associated with one Calling Search Space.

4.7.21.2 Fields

The `RoutePartitionCallingSearchSpaceMember` class has the following fields:

Table 4.103 Route Partition Calling Search Space Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.22 Route Skill Group Member

4.7.22.1 Description

The Route Skill Group Member defines a relationship between a Route and a Skill Group. It is a one-one relationship.

4.7.22.2 Fields

The RouteSkillGroupMember class has the following fields:

Table 4.104 Route Skill Group Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
No further fields.				

4.7.23 Service Enterprise Service Member

4.7.23.1 Description

The Service Enterprise Service Member defines a relationship between a Service and an Enterprise Service. It is a many-many relationship.

4.7.23.2 Fields

The ServiceEnterpriseServiceMember class has the following fields:

Table 4.105 Service Enterprise Service Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
No further fields.				

4.7.24 Service Media Routing Domain Member

4.7.24.1 Description

The Service Media Routing Domain Member defines a relationship between an Service and a Media Routing Domain. It is a many-one relationship and each Service can only be associated with one Media Routing Domain.

4.7.24.2 Fields

The ServiceMediaRoutingDomain class has the following fields:

Table 4.106 Service Media Routing Domain Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
No further fields.				

4.7.25 Service Peripheral Member

4.7.25.1 Description

The Service Peripheral Member defines a relationship between a Service and a Peripheral. It is a many-one relationship and each Service can only be associated with one Peripheral.

4.7.25.2 Fields

The ServicePeripheralMember class has the following fields:

Table 4.107 0Service Peripheral Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
No further fields.				

4.7.26 Skill Group Enterprise Skill Group Member

4.7.26.1 Description

The Skill Group Enterprise Skill Group Member defines a relationship between a Skill Group and an Enterprise Skill Group. It is a many-many relationship.

4.7.26.2 Fields

The SkillGroupEnterpriseSkillGroup class has the following fields:

Table 4.108 Department Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: DimensionMember				
No further fields.				

4.7.27 Skill Group Media Routing Domain Member

4.7.27.1 Description

The Skill Group Media Routing Domain Member defines a relationship between a Skill Group and a Media Routing Domain. It is a many-one relationship and each Skill Group can only be associated with one Media Routing Domain.

4.7.27.2 Fields

The `SkillGroupMediaRoutingDomainMember` class has the following fields:

Table 4.109 Skill Group Media Routing Domain Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.28 Skill Group Peripheral Member

4.7.28.1 Description

The Skill Group Peripheral Member defines a relationship between a Skill Group and a Peripheral. It is a many-one relationship and each Skill Group can only be associated with one Peripheral.

4.7.28.2 Fields

The `SkillGroupPeripheralMember` class has the following fields:

Table 4.110 Skill Group Peripheral Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
No further fields.				

4.7.29 Skill Group Service Member

4.7.29.1 Description

The Skill Group Service Member defines a relationship between a Skill Group and a Service. It is a many-one relationship and each Skill Group can only be associated with one Service.

4.7.29.2 Fields

The `SkillgroupServiceMember` class has the following fields:

Table 4.111 Skillgroup Service Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `DimensionMember`				
`PriorityLevel`	`Int` (n-v pair)	This is the priority level of the specified service for the specified skill group.	No	Yes

4.7.30 User Group Member

4.7.30.1 Description

The User Group Member defines a relationship between a User and a Group. It is a many-many relationship.

4.7.30.2 Fields

The `UserGroupMember` class has the following fields:

Table 4.112 User Group Member Fields

Element Name	Data Type	Description	Required?	Pkey?
Inherits From: `SystemMember`				
No further fields.				

4.8 Resource Management Web Service APIs

4.8.1 Create

4.8.1.1 Description

The `Create()` API creates a supported provisionable item type. For example, use this API to create an Agent.

In general the provisioning operation is synchronous for system resources such as users and folders and asynchronous for remote resources such as agents. Remote resources are queued for provisioning onto the underlying equipment, based on the effective from date field. They are provisioned when they become effective and when there is bandwidth on the remote equipment.

4.8.1.2 Parameters

The `Create()` API expects the following parameters:

Table 4.113 Parameters for Create API

SOAP Element Name	Data Type	Description	Required?
`resources`	`Resource[]`	An array of resources to be provisioned to the Contact Center environment (see section 4.3.1 "The Resource Hierarchy"). ***Note.*** *The Identity field should be set to -1 for each new resource being created.*	Yes

4.8.1.3 Return Type

The `Create()` API returns the following objects:

Table 4.114 Return Type for Create API

Data Type	Description	Required?
`RequestResult[]`	An array of results from the resources to be created (see section 4.3.4.5 "Request Result").	Yes

4.8.1.4 REST Protocol

Table 4.115 REST Protocol URI for Create API

Data Type	Description

URL	`https://<server>:8085/ResourceManagement/rest/resources/<resource type>s`
	Or, to create multiple resources of different types
	`https://<server>:8085/ResourceManagement/rest/resources`

HTTP Method	`POST`
Input/Output format	XML or JSON

Examples	`https://APPSRV01:8085/ResourceManagement/rest/Resources/folders`
	
	`https://APPSRV01:8085/ResourceManagement/rest/Resources`

4.8.1.5 Sequence Diagram

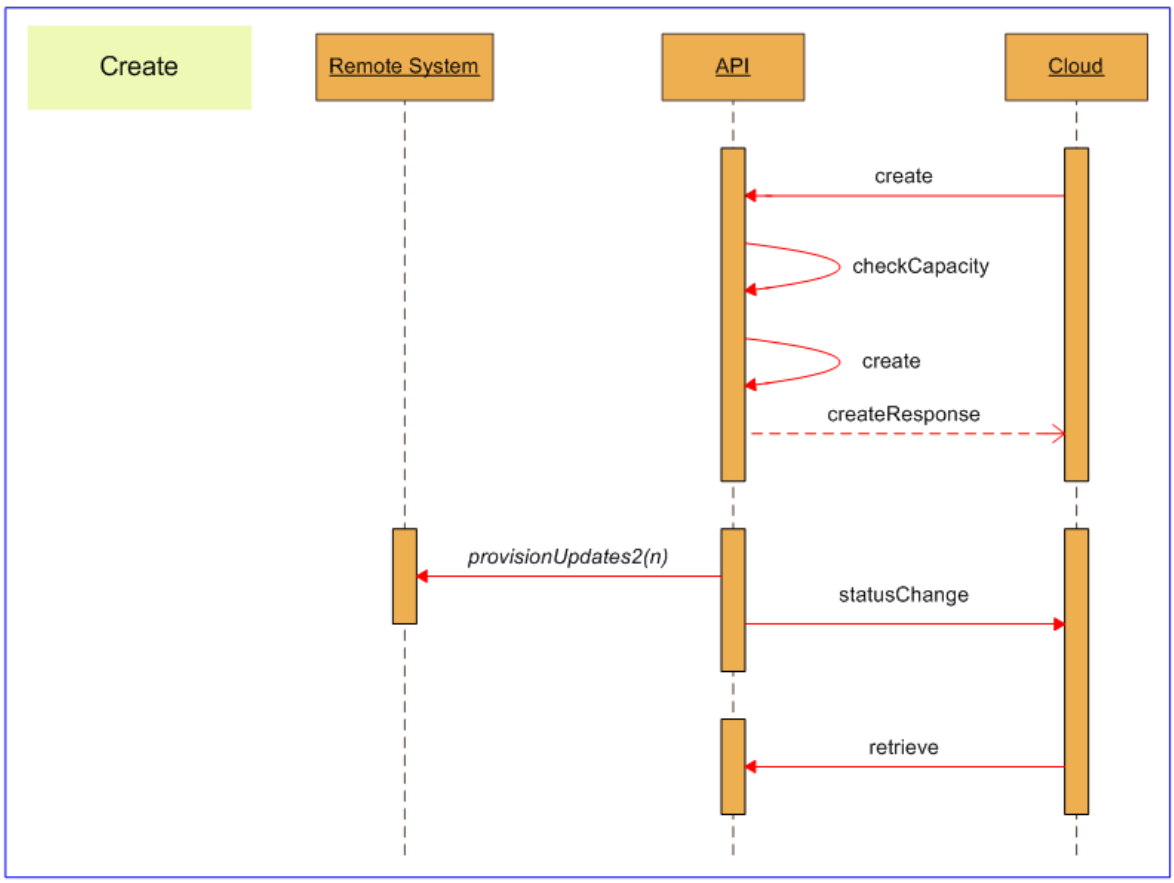


Figure 4.29 Sequence Diagram for Resource Management Create API

4.8.1.6 Creating Items with Pkey Maps

When you create a resource or membership that is mapped to more than one equipment instance, you need to specify a pkey map in `EquipmentMapping` for each equipment instance. To do this:

- In `Fields` set the item fields that are not equipment-specific, or are common to all equipment instances. These values will be applied to all equipment instances.

- In `EquipmentMapping` create a pkey map resource for each equipment instance (identified by a unique value for `ClusterResourceId`) and populate it with the fields that have equipment-specific values. These values will only apply to the specified equipment instance and, if necessary, will override any common values that were specified in `Fields`.

For example, for an Agent on multiple equipment instances:

- `AgentDesktopUrn` and `PersonUrn` are not specific to any equipment, so will be always be included in `Fields` of the `Agent` resource.
- A specific Agent may be a Supervisor on all the equipment instances, so the `Supervisor` flag can be included in `Fields` of the `Agent` resource if required.
- A specific Agent will need a `AgentPkey` resource in `EquipmentMapping` for each equipment instance, with `ClusterResourceId` set to the unique value for that equipment instance.
- A specific Agent may have one name on one equipment instance and a different name on another equipment instance, so `InternalName` can be set separately for each `AgentPkey` resource in `EquipmentMapping` if required.

Additionally, if you want to create several similar items on different equipment instances, you can specify a comma-separated list of equipment instances in `Fields.MappedClusterResources`. This will automatically create a pkey map resource in `EquipmentMapping` for each item in the list, and populate `EquipmentMapping.ClusterResourceId` with the corresponding value from the list. The values for the other fields will be as specified in `Fields`. If required, you can specify a comma-separated list of equipment instances in `Fields.MappedClusterResources` as well as creating one or more pkey map resources for specific equipment instances with specified `ClusterResourceId` values in `EquipmentMapping`.

For example, you may create an Agent with the same details on two equipment instances:

- Set the `Fields` of the `Agent` resource to the common values to be applied to both equipment instances.
- Set `Fields.MappedClusterResources` to the unique identifiers for the two equipment instances, separated by a comma.

For a SOAP example that creates an Agent mapped to two equipment instances, see section 4.9.38 "Create Agent with Pkey Maps, Method 1 (Explicit Pkey Maps) ". This example uses pkey maps to specify two separate equipment mappings.

For a SOAP example that creates an Agent Team mapped to two equipment instances, see section 4.9.39 "Create Agent Team with Pkey Maps, Method 2 (Implicit Pkey Maps)". This example specifies a comma-separated list of equipment instances in `Fields.MappedClusterResources`.

For a SOAP example that creates an Agent Agent Team membership see section 4.9.42 "Create Agent To Agent Team Membership (Two Remote Equipment Mappings)". This example uses pkey maps to link the agent on each equipment instance with the corresponding agent team on the same equipment instance.

4.8.2 Update

4.8.2.1 Description

The `Update()` API updates supported provisionable item types. For example, use this API to add a Skill Group membership to an Agent.

4.8.2.2 Parameters

The `Update()` API expects the following parameters:

Table 4.116 Parameters for Update API

SOAP Element Name	Data Type	Description	Required?
`resources`	`Resource []`	An array of resources to be provisioned to the Contact Center environment (see section 4.3.1 "The Resource Hierarchy").	Yes

4.8.2.3 Return Type

The `Update()` API returns the following objects.

Table 4.117 Return Type for Update API

Data Type	Description	Required?
`RequestResult []`	An array of result from the resources to be updated (see section 4.3.4.5 "Request Result").	Yes

4.8.2.4 REST Protocol

Table 4.118 REST Protocol URI for Update API

Data Type	Description

URL	`https://<server>:8085/ResourceManagement/rest/resources/<resource type>s/<id>`
	Or, to update multiple resources of different types
	`https://<server>:8085/ResourceManagement/rest/resources/<resourceKeys>`

HTTP Method	`PUT`
Input/ Output format	XML or JSON
Example	`https://APPSRV01:8085/ResourceManagement/rest/Resources/agents/315553`

4.8.2.5 Sequence Diagram

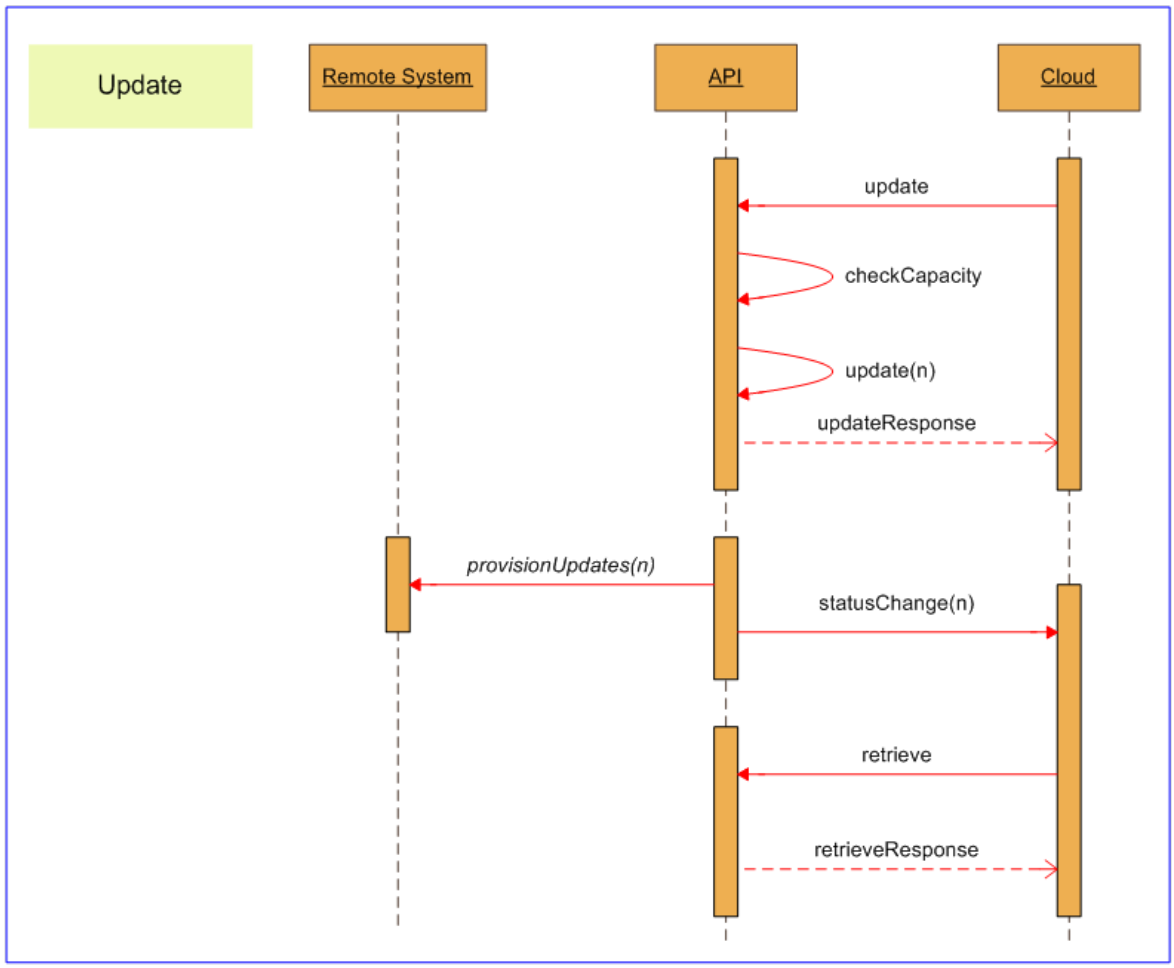


Figure 4.30 Sequence Diagram for Resource Management Update API

4.8.2.6 Updating Items with Pkey Maps

When you update a resource or membership that is mapped to more than one equipment instance, to access equipment-specific information, you need to specify a pkey map in `EquipmentMapping` for the equipment instance.

Specify the pkey map in the same way as described in section 4.8.1.6 "Creating Items with Pkey Maps".

Note
You can use `Update()` to add a new pkey map, but you cannot use `Update()` to delete an existing pkey map. To delete an existing pkey map from an item, use `Delete()` on the corresponding pkey item.

For a SOAP example that updates an Agent mapped to two equipment instances, where only the details for one equipment instance are to be updated, see section 4.9.39 "Create Agent Team with Pkey Maps, Method 2 (Implicit Pkey Maps)". This example specifies the equipment instance in `Fields.ClusterResourceId`.

4.8.3 Delete

4.8.3.1 Description

The `Delete()` API deletes supported provisionable item types. For example, use this API to delete a Skill Group.

4.8.3.2 Parameters

The `Delete()` API expects the following parameters:

Table 4.119 Parameters for Delete API

SOAP Element Name	Data Type	Description	Required?
`resourceKeys`	`ResourceKey[]`	An array of resource identities for the items to be deleted (see section 4.3.4.3 "Resource Key").	Yes

4.8.3.3 Return Type

The `Delete()` API returns the following objects:

Table 4.120 Return Type for Delete API

Data Type	Description	Required?
`RequestResult[]`	An array of results from the resources to be deleted (see section 4.3.4.5 "Request Result").	Yes

4.8.3.4 REST Protocol

Table 4.121 REST Protocol URI for Delete API

Data Type	Description

URL	`https://<server>:8085/ResourceManagement/rest/resources/<resource type>s/<id>`
	Or, to delete multiple resources of different types
	`https://<server>:8085/ResourceManagement/rest/resources/<resourcekeys>`

HTTP Method	`DELETE`
Input/ Output format	XML or JSON
Example	`https://APPSRV01:8085/ResourceManagement/rest/Resources/agents/315553`

4.8.3.5 Sequence Diagram

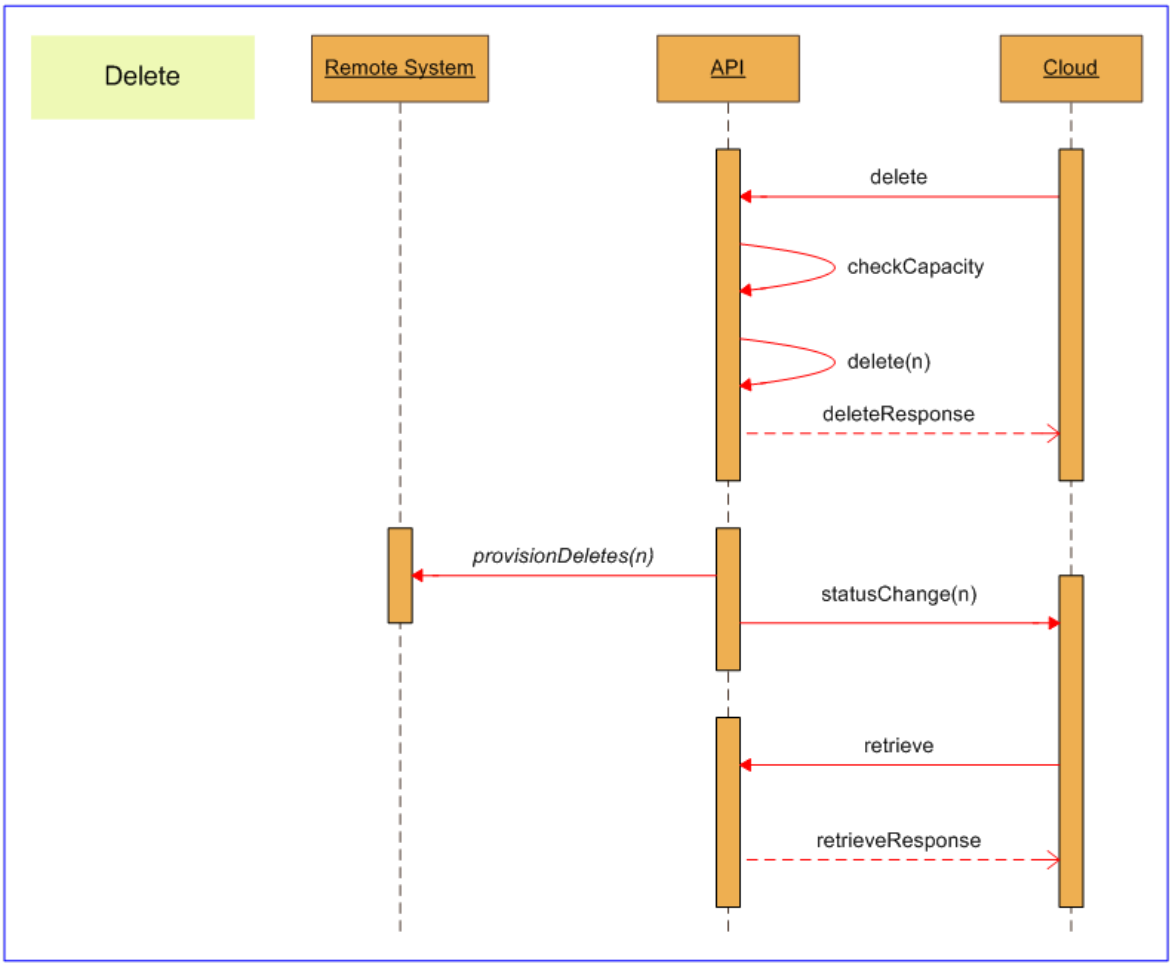


Figure 4.31 Sequence Diagram for Resource Management Delete API

4.8.3.6 Deleting Items with Pkey Maps

When you delete a resource or membership that is mapped to more than one equipment instance, all pkey maps are deleted with the item.

If you want to delete a resource or membership mapping to a specific equipment instance without deleting the item itself, use the `Update()` method on the parent item, and set the field `status` to `deleted`.

Note
You cannot delete the last pkey map for a resource or membership. You must delete the item itself.

4.8.4 Save

4.8.4.1 Description

The `Save()` API allows you to create, update and delete multiple item and types in a single API call.

Use this API

- for efficiency to avoid provisioning separate create and delete operations for the same item, (for example, when reskilling many agents at the start of a shift, which requires a delete followed by an add)
- when a two different operations have to be performed concurrently to preserve data integrity (for example, when creating a new Precision Queue Step in a Precision Queue and updating an existing Precision Queue Step in the Precision Queue at the same time).

4.8.4.2 Parameters

The `Save()` API expects the following parameters:

Table 4.122 Parameters for Save API

SOAP Element Name	Data Type	Description	Required?
`resourcesToAdd`	`Resource[]`	An array of resources to be created or updated. Must be supplied, but may be empty. ***Note.*** *The Identity field should be set to -1 for each new resource being created.*	Yes
`resourcesToDelete`	`Resource[]`	An array of resources to be deleted. Must be supplied, but may be empty.	Yes

4.8.4.3 Return Type

The `Save()` API returns the following objects:

Table 4.123 Return Type for Create API

Data Type	Description	Required?
`RequestResult []`	An array of results from the resources to be created, updated or deleted.	Yes

4.8.4.4 REST Protocol

Table 4.124 REST Protocol URI for Save API

Data Type	Description

URL	If request contains all the same type of resource:
	`https://<server>:8085/ResourceManagement/rest/resources/members/{resourceMemberTypeCollection}`
	If request contains more than one type of resource:
	`https://<server>:8085/ResourceManagement/rest/resources/members`

HTTP Method	`POST`
Input/ Output format	XML or JSON
Example	`https://APPSRV01:8085/ResourceManagement/rest/Resources/members/folders`

4.8.4.5 Sequence Diagram

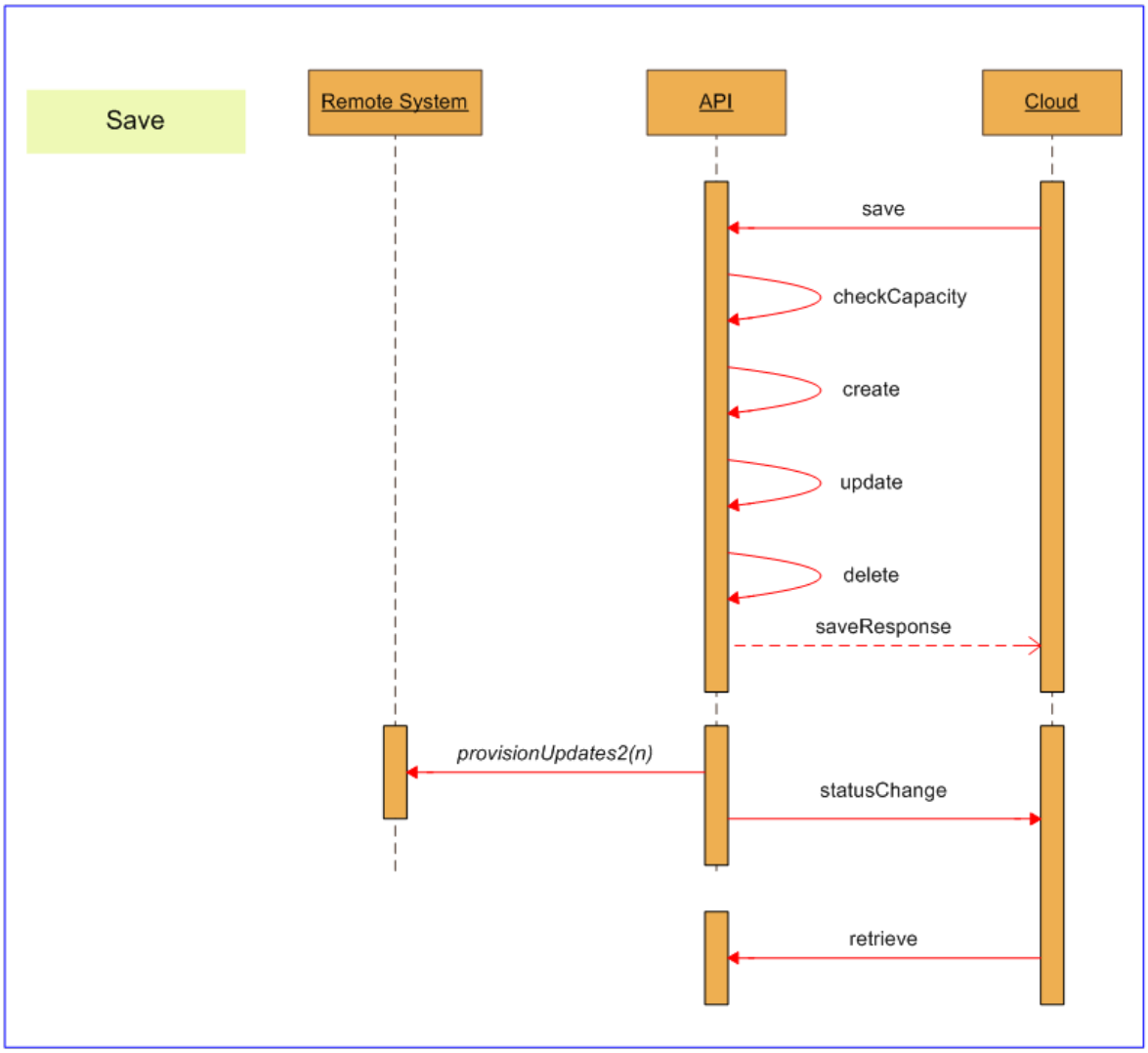


Figure 4.32 Sequence Diagram for Resource Management Save API

4.8.4.6 Saving Items with Pkey Maps

When you save a resource or membership that is mapped to more than one equipment instance, the pkey map is handled in the same way as for the underlying create, update or delete operation.

4.8.5 Move

4.8.5.1 Description

The `Move()` API moves one or more resource items to the specified folder from their current folder location or locations.

For example, you can use this API to move multiple Agents to a new folder in a single action, instead of using `Update()` and needing to specify the destination folder separately for each Agent.

4.8.5.2 Parameters

The `Move()` API expects the following parameters:

Table 4.125 Parameters for Move API

SOAP Element Name	Data Type	Description	Required?
`destinationFolderId`	`Int`	The identifier of the folder where the resource items will be moved to.	Yes
`resourceKeys`	`ResourceKey []`	An array of resource identities for the items to be moved (see section 4.3.4.3 "Resource Key").	Yes

4.8.5.3 Return Type

The `Move()` API returns the following objects:

Table 4.126 Return Type for Move API

Data Type	Description	Required?
`RequestResult []`	An array of results from the resources to be moved (see section 4.3.4.5 "Request Result").	Yes

4.8.5.4 REST Protocol

Table 4.127 REST Protocol URI for Move API

Data Type	Description
URL	`https://<server>:8085/ResourceManagement/rest/resources/destination/{destinationFolderid}`
HTTP Method	`POST`
Input/ Output format	XML or JSON
Example	`https://APPSRV01:8085/ResourceManagement/rest/Resources/destination/<destinationfolderid>`

4.8.5.5 Sequence Diagram

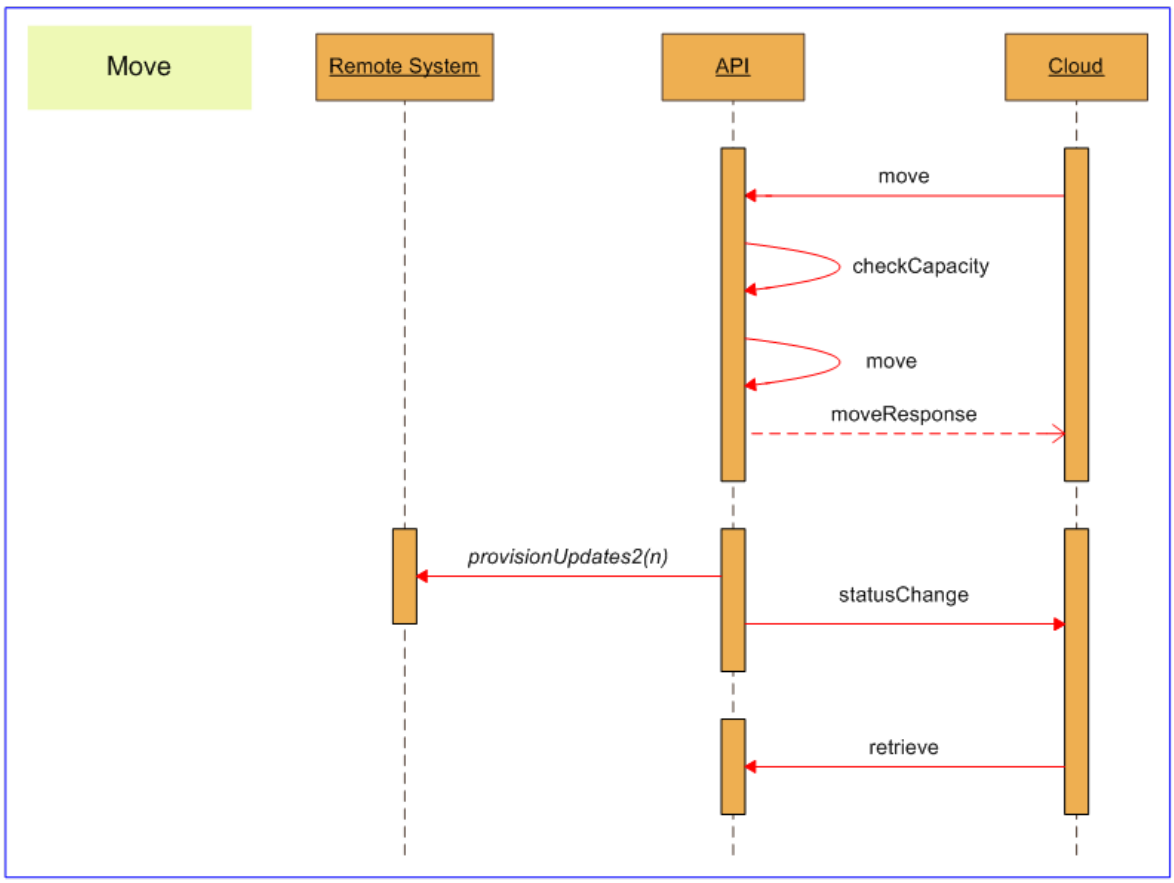


Figure 4.33 Sequence Diagram for Resource Management Move API

4.8.5.6 Moving Items with Pkey Maps

When you move a resource that is mapped to more than one equipment instance, the pkey map is handled in the same way as for the underlying update operation.

4.8.6 Retrieve

4.8.6.1 Description

The `Retrieve()` API returns a collection of Resource objects that relate to the supplied `ResourceKey` objects.

Use this API to retrieve collections of specific resources for which information is to be displayed, for example, when a Skill Group is selected from a list of Skill Groups and the properties specific to that Skill Group are to be displayed.

4.8.6.2 Parameters

The `Retrieve()` API expects the following parameters:

Table 4.128 Parameters for Retrieve API

SOAP Element Name	Data Type	Description	Required?
`resourceKeys`	`ResourceKey []`	An array of resource identities for the items to be retrieved (seesection 4.3.4.3 "Resource Key")	Yes

4.8.6.3 Return Type

The `Retrieve()` API returns the following objects:

Table 4.129 Return Type for Retrieve API

Data Type	Description	Required?
`Resource []`	An array of resources of the items relating to the identities passed in. Items will be returned in the order of their corresponding identities as passed into the method (see section 4.3.3 "Resource Hierarchy Classes").	Yes

4.8.6.4 REST Protocol

Table 4.130 REST Protocol URI for Retrieve API

Data Type	Description

URL	`https://<server>:8085/ResourceManagement/rest/resources/<resource type>s/<id>`
	Or, to retrieve multiple resources of different types,
	`https://<server>:8085/ResourceManagement/rest/resources/<resourcekeyCollection>`

HTTP Method	`GET`

Data Type	Description
Input/ Output format	XML or JSON
Examples	`https://APPSRV01:8085/ResourceManagement/rest/Resources/agents/315553` `https://APPSRV01:8085/ResourceManagement/rest/Resources/1234,IT_AGENT|2345,IT_AGENT_TEAM`

4.8.6.5 Sequence Diagram

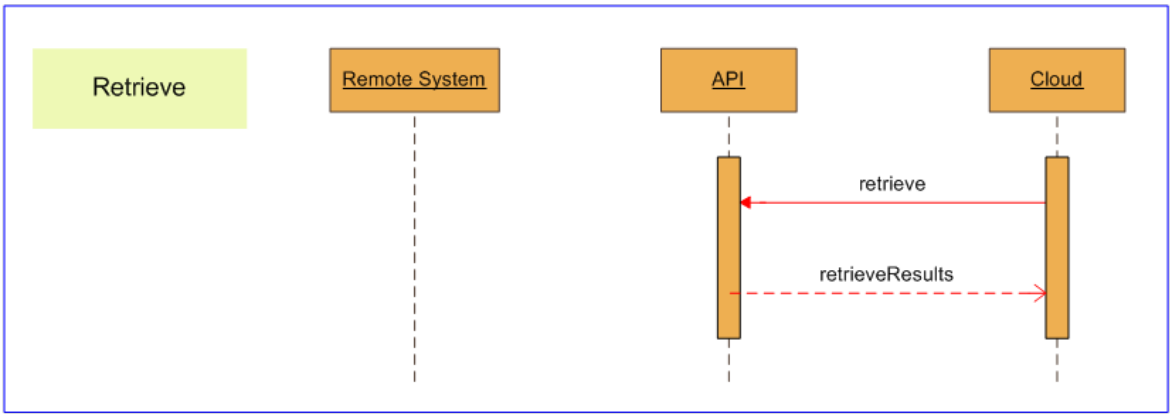


Figure 4.34 Sequence Diagram for Resource Management Retrieve API

4.8.6.6 Retrieving Items with Pkey Maps

When you retrieve a resource or membership that is mapped to more than one equipment instance, the following applies:

- `Fields` contains the details for all fields that are not equipment-specific.
- `Fields` also contains the equipment-specific details for the earliest equipment mapping that is still active.
- `EquipmentMapping` contains a pkey map for each of the other equipment instances. This pkey map only contains the fields that are specific to that equipment instance.

This means that retrieved resources and memberships can be accessed successfully by both legacy clients that do not expect pkey map details and by clients that are aware of the pkey map resource type.

4.8.7 Search

4.8.7.1 Description

The `Search()` API retrieves items based on item type, name, relationships etc. The client can specify search terms to control the search. The client can also specify a collection of items to be excluded from the query results, even if they match the specified search terms.

4.8.7.2 Parameters

The `Search()` API expects the following parameters:

Table 4.131 Parameters for Search API

SOAP Element Name	Data Type	Description	Required?
`queryString`	`String`	The search string containing the search terms that specify the items to be retrieved. (see Table 4.132).	Yes
`excludeFilter`	`ResourceKey []`	A collection of `ResourceKey` objects for items to be excluded from the search results (see section 4.3.2 "Common Fields and Classes").	No

4.8.7.3 Search String Syntax

The search string is constructed from one or more of search terms. Table 4.132 lists the available search terms, using the following conventions:

- Text enclosed by `{}` indicates an expression to be replaced with your own value, for example `{Id}` must be replaced with an item id.
- An ellipsis (`...`) indicates that the preceding term or phrase may be repeated as many times as required. For example, the search term syntax

  ```
  item:{IdOrName},{IdOrName}...
  ```

  indicates that the `item` search term requires one value that is an id or name, followed optionally by as many more id or name values as are required, each separated by a comma.
- Double quotes (“) must be used to delimit strings that contain embedded spaces.

Search terms can be combined or negated to give a powerful and flexible mechanism for retrieving specific items.

Negation

To negate a search term, add a "-" character before the search term. Any items that match the search term following the "-" character will be excluded from the result set.

For example, the search term

```
folder:/cicm -text:"HIDDEN ITEM"
```

gives a result set containing all items in the **/cicm** folder except those containing "HIDDEN ITEM" in their name or description.

Combined Search Expressions

Search terms can be combined using AND and OR operators to build up a search expression. By default, specifying two search terms with a space between the terms is interpreted as an AND operation, and only those results that meet both criteria are returned. OR operators can be applied to expressions but cannot be used to create nested expressions.

For example, the search term

```
folder:/cicm name:"Bob" OR folder:/cicm2 name:"Dave"
```

will return an item called “Dave” in the **/cicm2** folder and an item called ”Bob” in the **/cicm** folder but will not return an item called” Dave” in the **/cicm** folder.

Global Search Terms

Global search terms can be defined which apply to the entire search string. Global search terms are marked with a "*" before the search term.

For example, the search string

```
*folder:/cicm name:"Bob" OR name:"Dave"
```

applies the `*folder:/cicm` search filter to the entire search string, and will return an item ”Dave” in the **/cicm** folder and an item ”Bob” in the **/cicm** folder but will not return an item ”Dave” in the **/cicm2** folder.

Modifier Search Terms

The following search terms are modifiers, and are used to modify the result set returned by the other search terms. They cannot be used on their own:

- `count`
- `effective`
- `effectiverange`
- `maxmodified`
- `offset`

- owner
- sort

Search Performance

The following search terms are efficient, and can be used to restrict the search results set before applying any specific property-based searches (which are slower):

- deleted
- effective
- effectiverange
- enabled
- hidden
- latest
- modified
- status
- system
- type

For best search performance, where possible, use the search terms in the list above instead of using a property search on the same field, which will be slower. The following search terms use property searches:

- property
- childof (if {MemberOptions} is specified)
- parentof (if {MemberOptions} is specified).

For example, the search string

```
deleted:false
type:”Agent Agent Team Member”
childof:”Agent Team”,1234
```

will be more efficient than

```
childof:”Agent Team”,1234
[Type=”Agent Agent Team Member”;Deleted=false]
```

although both search strings will return all non-deleted Agent Agent Team Member items for the Agent Team “1234”.

Sorting Search Results

The sort search term allows you to specify a field or fields to use to sort the returned results.

As well as the fields available for a particular item type, some items also have associated name fields that can also be used to sort the results. Associated name fields translate item fields that are ids, GUIDs or URNs into name strings. They allow search results to be ordered alphabetically by name, rather than by internal identifier, without requiring extra API calls to reformat the results.

For example, the search string

```
folder:/cicm/mydept
type:"Agent Team"
sort:DialedNumberUrn
```

returns all the agent teams in the specified folder, sorted by the URN of the default dialed number, whereas the search string

```
folder:/cicm/mydept
type:"Agent Team"
sort:ParentDialedNumberDisplayName
```

returns all the agent teams in the specified folder, sorted by the dialed number display name. This allows the web application to display the returned results in a more meaningful order without additional processing.

The associated name fields available for each item are specified with the item specification.

Search Latency

The Search API uses enterprise level caching, which may lead to a delay between create, update or delete operations and their state being reflected in the search results. It is recommend that clients maintain a list of changes that may have not been propagated to the cache at the point the search is called. The SOAP client can use the subscriptions mechanism (see Chapter 5 "Subscriptions Web Service") to subscribe for notifications when items are provisioned, and remove the items from the list as the notifications are received.

Search Terms

Table 4.132 Search Terms for Search API

Search Term	Description	Syntax and Examples
`count`	Return the number of items that would be returned by the specified search terms.	**Syntax** `count:` **Examples** `cstext:"SkillGroup For Spanish" count:` returns the number of resources that match the `cstext` search term "SkillGroup For Spanish".

`cstext`	Case-sensitive search for items with the specified text in the `Name`, `InternalName` or `Description` fields.
	This search is much slower than the `key` search but provides a finer grained search.
		Syntax
		One of:
		`cstext:{SearchString}`
		`cstext:{SearchString},{SearchField}`
		`{SearchString}` specifies the string to search for, and the optional `{SearchField}` is one of:
		`n` search for specified text in `Name` field only
		`i` search for specified text in `InternalName` field only
		`d` search for specified text in `Description` field only
		Examples
		`cstext:"Spanish"` returns items where `Name`, `InternalName` or `Description` contains “Spanish” (but not “SPANISH”, or “spanish” etc).
		`cstext:"SkillGroup For Spanish"`.
		`cstext: “David”,i` returns items where the internal name contains “David”, but not items where the internal name contains “david” or “DAVID”.

Search Term	Description	Syntax and Examples
childof	Restrict the search to children of the specified parent. Both a type and an id are needed to ensure the parent is identified uniquely. If the type is not supplied, then the children of all items with the specified id will be returned. If the id is not supplied then the children of all items of the specified type will be returned. Where possible, specify both a type and an id as this makes the search more efficient.	**Syntax** One of: childof:{Type} childof:{Id} childof:{Type},{Id} childof:{Type}[{MemberOptions}] childof:{Id}[{MemberOptions}] childof:{Type},{Id}[{MemberOptions}] The optional {MemberOptions} is enclosed in square brackets and specifies one or more property filters to apply to the membership children of the specified parent. Each membership property filter in {MemberOptions} is of the form {PropertyName}{Comparison}{PropertyValue}, where {Comparison} is one of: ! is not equal to > is greater than < is less than } is greater than or equal to { is less than or equal to If {MemberOptions} contains more than one membership property filter, each filter must be separated by a semi-colon (;). For example, [Type="Agent Skillgroup Member"; DefaultMember=true], specifies a membership property filter for Agent Skillgroup Members where the membership represents the default skillgroup for that agent. **Note**: The {MemberOptions} property filter only affects child membership items. It does not affect child resource items, and all child resource items of the specified parent will be returned, even with a {MemberOptions} filter specified. If you want to restrict the result set further, include one of the other search terms, for example, type: or property:. **Examples** childof:"Skill Group",1234 returns all child items associated with the Skill Group with id "1234". childof:1234 returns all child items associated with any item with id "1234". childof:Peripheral returns all child items

Search Term	Description	Syntax and Examples
		associated with all Peripherals. `childof:Peripheral,5678 type:Agent deleted:false enabled:true` searches the Peripheral with id "5678" and returns all enabled, non-deleted Agents associated with this Peripheral. `childof:"Skillgroup",5555 [Type="Agent Skillgroup Member"; DefaultMember=true]` searches Skillgroup "5555" and returns all associated Agent Skillgroup Member items where the membership represents the default skillgroup for an agent.
`deleted`	Specify whether to include deleted items in the search. By default, the search results will include both deleted and non-deleted items.	**Syntax** `deleted:{Flag}` `{Flag}` is one of `false`, `true`, `0` (false) or `1` (true) **Examples** `deleted:true` returns only deleted items. `deleted:false` returns only non-deleted items. `deleted:0` returns only non-deleted items.

`effective`	Restrict the search to items that were effective at the specified date and time.	**Syntax**
		`effective:{DateTime}`
		`{DateTime}` is specified as `yyyy-mm-ddThh:mm:ss`
		Examples
		`effective:"2009-09-20T23:00:00"` returns only items that were effective at 2300 on 20th September 2009.

`effective range`	Restrict the search to items that were effective during the supplied date and time range.	**Syntax**
		`effectiverange:{StartDateTime}, {EndDateTime}`
		`{StartDateTime}` and `{EndDateTime}` are specified as `yyyy-mm-ddThh:mm:ss`
		Examples
		`effectiverange:"2009-09-20T23:00:00", "2009-09-21T23:00:00"` returns only items that were effective between 2300 on 20th September 2009 and 2300 on 21th September 2009.

Search Term	Description	Syntax and Examples
`enabled`	Specify whether to include enabled items in the search. By default, the search results will include both enabled and disabled items.	**Syntax** `enabled:{Flag}` `{Flag}` is one of `false`, `true`, `0` (false) or `1` (true) **Examples** `enabled:true` returns only enabled items. `enabled:false` returns only disabled items. `enabled:1` returns only enabled items.

Search Term	Description	Syntax and Examples

`folder`	Restrict the search to items in the specified folder, and optionally, subfolders or parent folders.
	This search filter only returns items that are in the folder tree. It will return remote resources and system resources (including other folders) but not member items.
		Syntax
		One of:
		`folder:{PathOrId}`
		`folder:{PathOrId}{Options}`
		The optional `{Options}` is one of
		`*` Search folder and subfolders (one level)
		`**` Search folder and subfolders (all levels)
		`/**` Search subfolders (all levels)
		`^` Search parent folders (all levels)
		If `{Options}` is not specified, then the search is restricted to the specified folder.
		Examples
		`folder:/cicm` returns items (including subfolders) in **/cicm.**
		`folder:/cicm/*` returns items (including subfolders) in the subfolders of **/cicm.**
		`folder:/cicm*` returns items (including subfolders) in **/cicm** and also items (including subfolders) in immediate subfolders of **/cicm.**
		`folder:/cicm**` returns items (including subfolders) in **/cicm** and also items (including subfolders) in all levels of subfolders of **/cicm.**
		`folder:/cicm/**` returns items (including subfolders) in all levels of subfolders of **/cicm.**
		`folder:/cicm/mydept^` returns items (including subfolders) at the same level as **/cicm/mydept**, and all levels above, excluding **/cicm/mydept** itself.
		`folder:{2833BE91-68D3-45E7-94BD-A7F2BC139167}` returns items in the folder with the specified id.
		`folder:{2833BE91-68D3-45E7-94BD-A7F2BC139167}/*` returns items in the subfolders of the folder with the specified id.

`hidden`	Specify whether to include hidden items in the search. By default, the search results will include both hidden and non-hidden (visible) items.	**Syntax** `hidden:{Flag}` `{Flag}` is one of `false`, `true`, `0` (false) or `1` (true) **Examples** `hidden:true` returns only hidden items. `hidden:0` returns only visible items.

Search Term	Description	Syntax and Examples
`internalname`	Search for items by `InternalName` (which corresponds to the name of the item on the remote equipment, for example, *EnterpriseName* in Unified CCE).	**Syntax** `internalname:{Name}` **Examples** `internalname:Agent1` returns all resources with internal name “Agent1”. `internalname:"Agent 1234"` returns all resources with internal name of “Agent 1234”.

`item`	Search for items by id, `Name` or `InternalName`.
	Both a type and an id are needed to ensure the item is identified uniquely. If a `type:` search term is not specified as well, then all items with the specified id will be returned.
		Syntax
		One of:
		`item:{IdOrName}`
		`item:{IdOrName},{IdOrName}...`
		If more than one `{IdOrName}` is specified, then each one must be separated by a comma (,).
		Examples
		`item:1234` returns all items with an id, name or internal name of “1234”.
		`item:1234,7654` returns all items with an id, name or internal name of “1234” or “7654”.
		`item:1234,7654 type:Agent` returns all Agents with an id, name or internal name of “1234” or “7654”.
		`item:"1234,Bob1"` returns all items with a name or internal name of "1234,Bob1".

`key`	Search for items by the item key. This search term accepts a sing;e item or a list of items, which may be of different types.
	This search is very efficient.
		Syntax
		One of:
		`key:{Type},{Id}`
		`key:{Type},{Id}|{Type},{Id}...`
		If more than one `{Type},{Id}` pair is specified, then each `{Type},{Id}` pair must be separated by a vertical bar (|).
		Examples
		`key:Agent,9474|Peripheral,2917|MT_ITEM_TENANT_MEMBER,2927` returns the Agent with id 9474, the Peripheral with id 2917 and the `MT_ITEM_TENANT_MEMBER` with id 2927.

Search Term	Description	Syntax and Examples
`latest`	Specify whether to include current items and items with type-2 changes in the search. By default, the search results will include both current items and items with type-2 changes.	**Syntax** `latest:{Flag}` `{Flag}` is one of `false`, `true`, `0` (false) or `1` (true) **Examples** `latest:1` returns only current items. `latest:false` returns only items with type-2 changes.
`max`	Restrict the number of items returned. Combine this with `offset` to return paged results.	**Syntax** `max:{Count}` **Examples** `max:2000` returns only the first 2000 matching items.
`member childof`	Search for member objects of the specified type which link to the specified parent. **Note**. This search term is included for legacy use only. New code should use the `childof:` search term to specify the parent, together with the `type:` search term to specify the required member type.	**Syntax** `memberchildof:{ParentType},{ParentId} {MemberType}` **Examples** `memberchildof:"Agent Team",4567, "Agent Agent Team Member"` returns all Agent Agent Team Member objects which have Agent Team 4567 as the parent.
`member parentof`	Search for member objects of the specified type which link to the specified child. **Note**. This search term is included for legacy use only. New code should use the `parentof:` search term to specify the child, together with the `type:` search term to specify the required member type.	**Syntax** `memberparentof:{ChildType},{ChildId} {MemberType}` **Examples** `memberparentof:"Agent",3541, "Agent Agent Team Member"` returns all Agent Agent Team Member objects which have Agent `3541` as the child.

Search Term	Description	Syntax and Examples

`modified`	Restrict the search to items modified on or after the specified date time.	**Syntax**
		One of:
		`modified:{DateTime}`
		`modified:{DeltaTime}`
		`{DateTime}` is specified as `yyyy-mm-dd hh:mm:ss` and the alternative `{DeltaTime}` is one of: `-1second`, `-1minute`, `-1hour`, `-1day`, `-{n} seconds`, `-{n}minutes`, `-{n}hours`, `-{n}days`
		Examples
		`modified:"2009-09-20 23:00:00"` returns only items that were modified on or after 2300 on 20th September 2009.
		`modified:-1hour` returns only items that were modified within the last hour.
		`modified:-90seconds` returns only items that were modified within the last 90 seconds.
		`modified:-7days` returns only items that were modified within the last week.

`name`	Search for items by name, or if `Name` is blank, by `InternalName`. ***Note.*** *This search term cannot be used to find items where both the name and the internal name are blank. You can use the `property` search term to do this, although this may be slow as the property search is not an indexed search.*	**Syntax** `name:{Name}` **Examples** `name:Agent1` returns the agent with the name or internal name of "Agent1". `name:"Agent 1234"` returns the agent with the name or internal name of "Agent 1234".

`owner`	Restrict the items returned to those owned by the specified tenant.	**Syntax**
		One of:
		`owner:{IdOrTenant}`
		`owner:{IdOrTenant},{IdOrTenant}...`
		If more than one `{IdOrTenant}` is specified, then each one must be separated by a comma (,).
		Examples
		`owner:/Tenant1` returns only items owned by Tenant1.
		`owner:{24811131-76E9-4406-9F66-711FD8716955}, {2ADA5C58-839E-4EC8-901E-1692E224B132}` returns only items owned by the tenants with the specified ids.

Search Term	Description	Syntax and Examples
`offset`	Restrict the items returned to those after the specified position in the results array. Combine this with `max` to return paged results. If the specified `offset` is greater than the number of matching results, the results array is empty.	**Syntax** `offset:{OffsetCount}` **Examples** `offset:100 max: 50` returns only the 101st to 150th matching items.

Search Term	Description	Syntax and Examples

parentof	Restrict the search to parents of the specified child.
	Both a type and an id are needed to ensure the child is identified uniquely. If the type is not supplied, then the parents of all items with the specified id will be returned. If the id is not supplied then the parents of all items of the specified type will be returned.
	Where possible, specify both a type and an id as this makes the search more efficient.
		Syntax
		One of:
		`parentof:{Type}`
		`parentof:{Id}`
		`parentof:{Type},{Id}`
		`parentof:{Type}[{MemberOptions}]`
		`parentof:{Id}[{MemberOptions}]`
		`parentof:{Type},{Id}[{MemberOptions}]`
		The optional `{MemberOptions}` is enclosed in square brackets and specifies one or more property filters to apply to the membership parents of the specified child.
		Each membership property filter in `{MemberOptions}` is of the form `{PropertyName}{Comparison}{PropertyValue}`, where `{Comparison}` is one of:
		`!` is not equal to
		`>` is greater than
		`<` is less than
		`}` is greater than or equal to
		`{` is less than or equal to
		If `{MemberOptions}` contains more than one membership property filter, each property filter must be separated by a semi-colon (;).For example, `[Type="Agent Skillgroup Member"; DefaultMember=true]`, specifies a membership property filter for Agent Skillgroup Members where the membership represents the default skillgroup for that agent.
		Note: The `{MemberOptions}` property filter only affects parent membership items. It does not affect parent resource items, and all parent resource items of the specified child will be returned, even with a `{MemberOptions}` filter specified. If you want to restrict the result set further, include one of the other search terms, for example, `type:` or `property:`.
		Examples
		`parentof:Agent,1234` returns all parent items associated with the Agent with id "1234".
		`parentof:1234` returns all parent items associated with any item with the id "1234".
		`parentof:"Agent"` returns all parent items associated with any Agent.

Search Term	Description	Syntax and Examples
		parentof: ”Agent”,5555 [Type=“Agent Skill Group Member”; DefaultMember=true] searches the Agent with id “5555” and returns the parent Skill Group Member item associated with the Agent where the membership represents the default skillgroup for that Agent. parentof:”Agent”,3541 type:“Agent Team” returns all Agent Team items which have Agent 3541 as the child.
property	Search for items based on the value of one or more of their properties. This search is case sensitive.	**Syntax** property:{PropertyName}{Comparison}{PropertyValue} {Comparison} is one of: ! is not equal to > is greater than < is less than } is greater than or equal to { is less than or equal to **Examples** property:FirstName=Bob returns all items where the first name is “Bob”. property:LastName!Smith returns all items where the last name is not “Smith”. property:WaitTime>20 returns all items where the wait time is greater than 20 seconds. property:Timeout}20 returns all items where the timeout is greater than or equal to 20 seconds.

Search Term	Description	Syntax and Examples
`sort`	Sort the returned items by the specified property.	**Syntax** One of: `sort:{SortTerm}sort:{SortTerm}$${SortTerm}...` `{SortTerm}` is one of: • `{PropertyName}|{AssociatedPropertyName}` (an ascending sort on the specified property name or associated property name) • `{PropertyName}|{AssociatedPropertyName},ASC` (an ascending sort on the specified property name or associated property name) • `{PropertyName}|{AssociatedPropertyName},DESC` (a descending sort on the specified property name or associated property name) If more than one `{SortTerm}` is specified, then each one must be separated by a dollar sign ($). Note that `{propertyName}` and `{AssociatedPropertyName}` are case-sensitive. **Examples** `sort:InternalName` returns the matching items, sorted by internal name in ascending order. `sort: InternalName,DESC` returns the matching items, sorted by internal name, in descending order. `sort:Supervisor$InternalName,DESC` returns the matching items, sorted first by supervisor in ascending order, then by internal name, in reverse order. `sort:ParentFolderPath,ASC` returns the matching items sorted by the folder path name that corresponds to the `FolderId` of each item.
`status`	Restrict the search to items matching the specified status.	**Syntax** `status:{Status}` **Examples** `status:R` only returns items that are in the ready state. `Status:D` only returns items that are in the deleted state.

Search Term	Description	Syntax and Examples
system	Specify whether to include items that are owned by the system in the search. By default, the search results will include both system items and non-system items. Items that are owned by the system include the root folder, system search folders, and the users, groups and roles that are created when the system is first installed. These items cannot be edited or deleted.	**Syntax** system:{Flag} **Examples** system:true only returns items owned by the system. System:false only returns items not owned by the system. System:0 only returns items not owned by the system.
text	Search for items with the specified text in the Name, InternalName or Description fields. This search is not case-sensitive. This search is much slower than the key search but provides a finer grained search.	**Syntax** One of: text:{SearchString} text:{SearchString},{SearchField} {SearchString} specifies the string to search for, and the optional {SearchField} is one of n search for specified text in Name field only i search for specified text in InternalName field only d search for specified text in Description field only **Examples** text:"Spanish" returns items where Name, InternalName or Description contains "Spanish" (or "SPANISH", or "spanish" etc) text:"SkillGroup For Spanish" text: "David",i returns items where InternalName contains "David", and also items where InternalName contains "david" or "DAVID".
type	Restrict the search to items of the specified type.	**Syntax** type:{Type} **Examples** type:IT_AGENTtype:"Agent Team"type:Foldertype:Peripheral
types	Return the item types supported by Search(). **Tip**: Call Describe() for any of the returned item types to get the fields and data types for that item type.	**Syntax** types:

4.8.7.4 Return Type

The `Search()` API returns the following objects:

Table 4.133 Return Type for Search API

Data Type	Description	Required?
`Resource[]`	The results of the search as a collection of Resource objects (see section 4.3.1 "The Resource Hierarchy").	Yes

4.8.7.5 REST Protocol

Table 4.134 REST Protocol URI for Search API

Data Type	Description
URL	`https://<server>:8085/ResourceManagement/rest/resources?queryString=<query string>&excludeFilter=<exclude filter>`
HTTP Method	`GET`
Input/ Output format	XML or JSON
Example	`https://APPSRV01:8085/ResourceManagement/rest/resources?queryString=type%3aAgent+max%3a1+latest%3a1&excludeFilter=`

4.8.7.6 Sequence Diagram

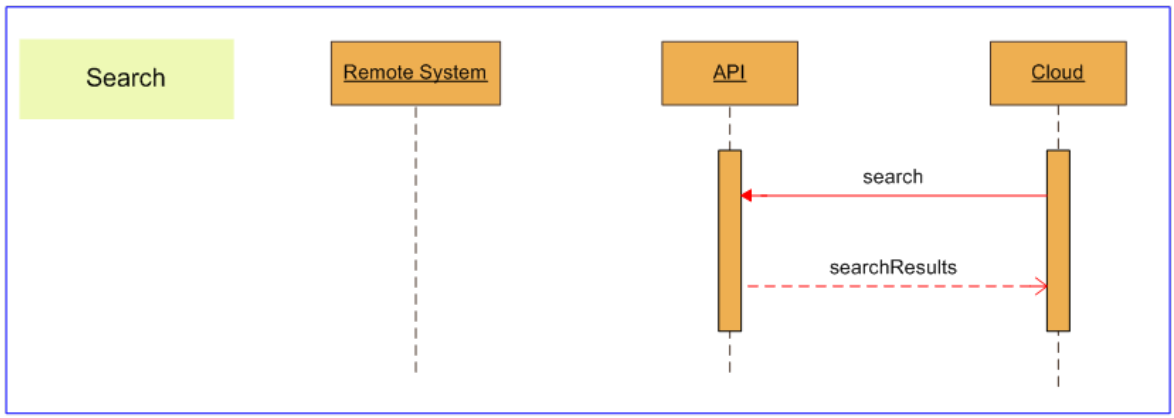


Figure 4.35 Sequence Diagram for Resource Management Search API

4.8.7.7 Searching and Items with Pkey Maps

If your search returns a resource or membership that is mapped to more than one equipment instance, `Fields` will include all fields for the earliest active equipment mapping. The pkey maps in `EquipmentMapping` contain the details for the other equipment instances.

For more details, see section 4.8.6.6 "Retrieving Items with Pkey Maps".

This means that search results can be accessed successfully by both legacy clients that do not expect pkey map details and by clients that are aware of the pkey map resource type.

Note

Pkey maps are implemented internally as memberships, so if you search for all member types, the pkey map memberships will be visible in the search results along with the resource-to-resource memberships.

4.8.8 Describe

4.8.8.1 Description

The `Describe()` API is used to return entity metadata for a specified resource type. This meta data can then be used for constraining population of fields when performing edit or create operations on objects of that type.

4.8.8.2 Parameters

The `Describe()` API expects the following parameters:

Table 4.135 Parameters for Describe API

SOAP Element Name	Data Type	Description	Required?
`resourceType`	`String []`	The resource types of the object or objects for which the field metadata is required. One or more of: the remote resource types listed in section 3.1 "Remote Resource Types", the system resource types listed in section 3.2 "System Resource Types", the resource member types listed in section 4.7.1.2 "Member Types and Identifiers", or the pkey map types listed in section 4.7.1.4 "Member Pkey Types".	Yes

4.8.8.3 Return Type

The `Describe()` API returns the following objects:

Table 4.136 Return Type for Describe API

Data Type	Description	Required?
`ResourceFieldMeta[]`	A collection of `ResourceFieldMeta` objects containing metadata for the resource type passed in. **Note**: For the remote resource types listed in section 3.1 "Remote Resource Types" and the member types listed in section 4.7.1.2 "Member Types and Identifiers", the returned metadata does not include fields that may be in a pkey map. To see these fields, you need to call `Describe()` and specify the corresponding resource or member pkey type.	Yes
`ClusterResourceType`	Future use. Defines the pkey elements that are relevant to the supplied cluster resource type.	No

4.8.8.4 REST Protocol

Table 4.137 REST Protocol URI for Describe API

Data Type	Description
URL	`https://<server>:8085/ResourceManagement/rest/resources/meta/<resource type>`
HTTP Method	`GET`

Data Type	Description
Input/ Output format	XML or JSON
Example	`https://APPSRV01:8085/ResourceManagement/rest/resources/meta/calling-search-space`

4.8.8.5 Sequence Diagram

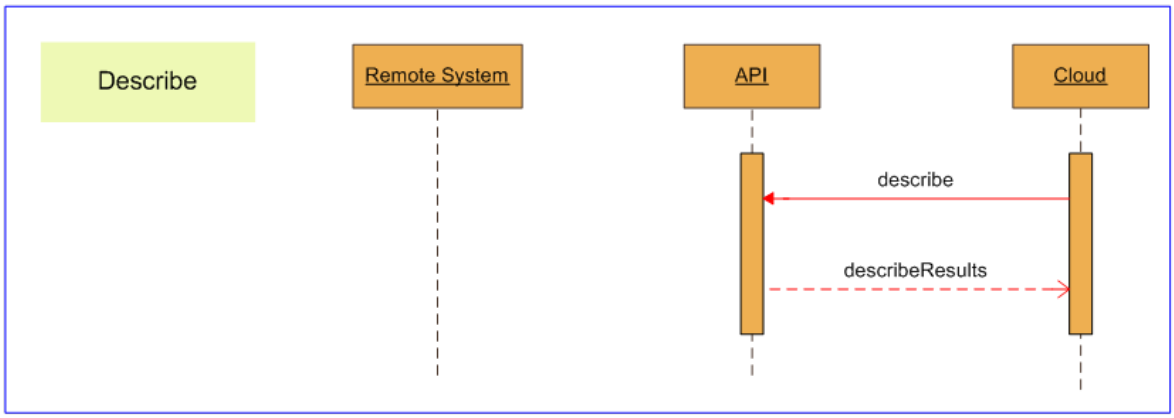


Figure 4.36 Sequence Diagram for Resource Management Describe API

4.8.8.6 Describing Items with Pkey Maps

If you describe a resource or membership that supports pkey maps, `Describe()` for the main item itself only returns the fields that are associated with the main item. To obtain the descriptions of the fields that may be returned on pkey maps, call `Describe()` for the corresponding resource or membership pkey type.

For example, to get the descriptions of all the fields for an Agent resource you will need to call `Describe (Agent, AgentPkey)`.

4.8.9 Audit

4.8.9.1 Description

The `Audit()` API returns the audit records associated with one or more resource entities. This allows the client to display the activity associated with a resource, for example, identifying the reasons for an error status.

4.8.9.2 Parameters

The `Audit()` API expects the following parameters:

Table 4.138 Parameters for Audit API

SOAP Element Name	Data Type	Description	Required?
`resourceKeys`	`ResourceKey[]`	An array of resource identities for the items whose audits are to be returned.	Yes
`eventOutcome`	`String`	A filter to reduce the returned audit records to those matching the outcome of the provisioning operation. One of: • **S**: Success • **F**: Failure • **N**: Not applicable • **U**: Unknown or empty string to match all audit records.	Yes
`fromDate`	`DateTime`	The start of the date range to filter the returned audit records. **Note**: for REST this is an 8601 encoded data string.	Yes
`toDate`	`DateTime`	The end of the date range to filter the returned audit records. **Note**: for REST this is an 8601 encoded data string.	Yes
`startIndex`	`Int`	Controls the pagination of audit records, specifies the index of the element at which to start.	Yes
`resultsPerPage`	`Int`	Controls the pagination of audit records, specifies the number of elements to retrieve.	Yes
`summary`	`Bool`	Whether the results are a summary or detailed results. If false, fills in the `AdditionalData` XML field in each of the returned audit records.	Yes

4.8.9.3 Return Type

The `Audit()` API returns the following object:

Table 4.139 Return Type for Audit API

Data Type	Description	Required?
`ResourceAuditResults`	The audit data for the resource id passed in (see section 4.3.4.9 "Resource Audit Results").	Yes

4.8.9.4 REST Protocol

Table 4.140 REST Protocol URI for Audit API

Data Type	Description

URL	`https://<server>:8085/ResourceManagement/rest/resources/<resource type>s/<id>/audits?fromDate=<from date>&toDate=<to date>&eventOutcome=S&summary=<true|false>`
	or, to audit multiple resources of different types,
	`https://<server>:8085/ResourceManagement/rest/resources/<resourceKeys>/audits?fromDate=<from date>&toDate=<to date>&eventOutcome=S&summary=<true|false>`

HTTP Method	`GET`
Input/ Output format	XML or JSON
Example	`https://APPSRV01:8085/ResourceManagement/rest/resources/agents/315553/audits?fromDate=2011-10-04T00%3a00%3a00Z&toDate=2012-06-06T10%3a43%3a57Z`

4.8.9.5 Sequence Diagram

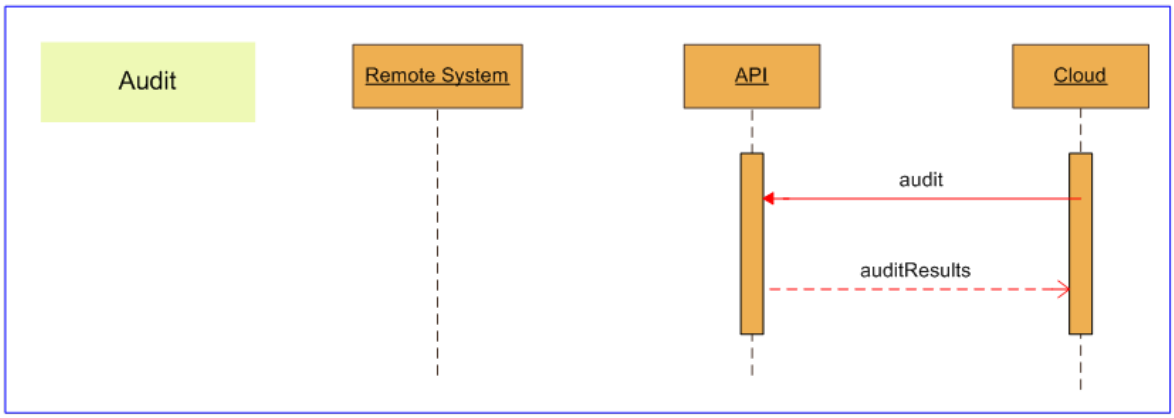


Figure 4.37 Sequence Diagram for Resource Management Audit API

4.8.9.6 Auditing Resources with Pkey Maps

There are no special considerations when auditing resources or memberships that support pkey maps, since `ResourceAudit` objects do not contain pkey maps.

4.8.10 Upload

4.8.10.1 Description

The `Upload()` API is a generic API that allows binary data to be associated with an existing resource and uploaded to the appropriate remote system. If there is existing binary data associated with the specified resource, this data is replaced with the new binary data. The type of content being uploaded and any additional data required to specify the upload are determined by the resource type.

The `Upload()` API encodes the additional data and binary data in separate parts of a **multipart/form-data** MIME message (see http://tools.ietf.org/html/rfc2388, link checked November 2013).

4.8.10.2 Parameters

The `Upload()` API expects the following parameters:

Table 4.141 Parameters for Upload API

SOAP Element Name	Data Type	Description	Required?
`ResourceKey`	`ResourceKey`	The resource key of the resource to associate the data with.	Yes
`ContentType`	`String`	The type of data being uploaded. Depends on the resource type (see table below).	Yes
`AdditionalData`	`AdditionalData`	The additional data required for the upload. Depends on the resource type (see Table 4.142). This parameter is encoded in Part 1 of a MIME message.	Yes
`Contents`	`Stream`	The binary data to be uploaded and associated with the specified resource. This parameter is encoded in Part 2 of a MIME message.	Yes

The supported resource types, and the associated content types and additional data required are:

Table 4.142 Supported Resource Types for Upload API

Resource Type	Content Type	Content Format	Additional Data		
			Name	Value	Required?
Media file	audio/wav	.wav file to be associated with the specified media file resource	Servers	Comma separated list of URNs of the media file servers where the media file is to be uploaded to. If not present, the media file will be uploaded to all media file servers associated with the specified media file resource.	No

4.8.10.3 Return Type

The `Upload()` API returns the following objects:

Table 4.143 Return Type for Upload API

Data Type	Description	Required?
RequestResult	The result of the upload.	Yes

4.8.10.4 REST Protocol

Table 4.144 REST Protocol URI for Upload API

Data Type	Description
URL	https://<server>:8085/ResourceManagement/rest/resources/<resource type>s/<id>/content?contentType=<contentType>
HTTP Method	PUT
Input/ Output format	XML or JSON
Example	https://<server>:8085/ResourceManagement/rest/resources/mediafiles/3565/content?contentType=audio/wav See Table 4.145 for an example MIME message for this URL.

Table 4.145 Example MIME Message for REST Protocol Upload API

Section	Example Content
Header	`Content-Type: multipart/form-data; boundary="----------8d08f053d6a90ad"` `Content-Length: 216829`

Part 1: Additional Data Format: XML or JSON	`----------8d08f053d6a90ad`
	`Content-Disposition: form-data; name="additionaldata"`
	`Content-Type: application/json; charset=utf-8`
	`{"Fields":[{"Name":"Servers","Value":"21,25,29"}]}`

Part 2: Content Format: File	`----------8d08f053d6a90ad`
	`Content-Disposition: form-data; name="content"; filename="datafile"`
	`Content-Type: audio/wav`
	(file to upload follows as binary data, up to message length defined in header section).

4.8.10.5 Sequence Diagram

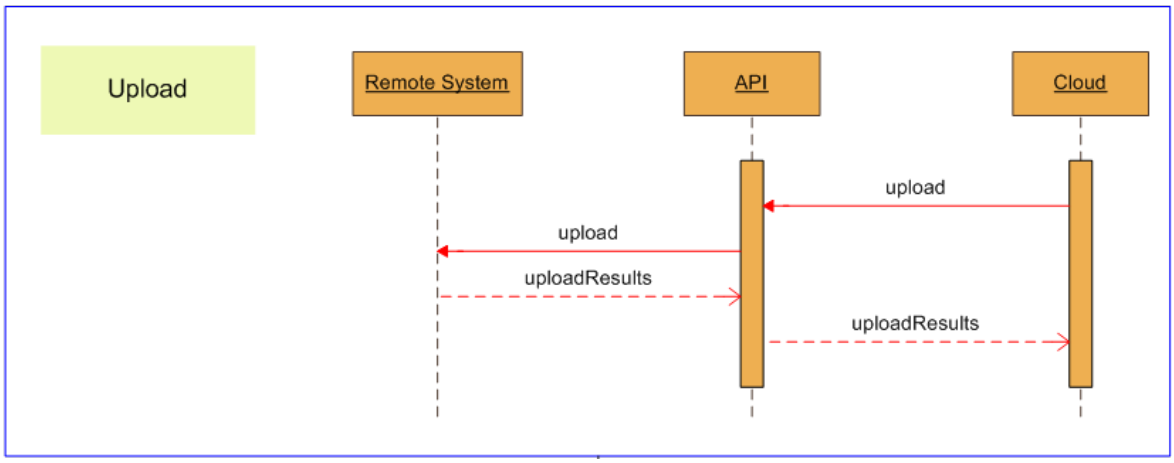


Figure 4.38 Sequence Diagram for Resource Management Upload API

4.8.11 Download

4.8.11.1 Description

The `Download()` API is a generic API that returns the binary data that has been associated with a resource. The type of content being returned is determined by the resource type.

4.8.11.2 Parameters

The `Download()` API expects the following parameters:

Table 4.146 Parameters for Download API

SOAP Element Name	Data Type	Description	Required?
`ResourceKey`	`ResourceKey`	The identity of the resource to return the binary data for.	Yes
`ContentType`	`String`	The type of binary data to be returned. Depends on the resource type (see Table 4.147).	Yes

The supported resource types, and the associated content types are:

Table 4.147 Supported Resource Types for Download

Resource Type	Content Type	Content Format
Media file	`audio/wav`	.wav file associated with the specified media file resource.

4.8.11.3 Return Type

The `Download()` API returns the following objects:

Table 4.148 Return Type for Download API

Data Type	Description	Required?
`Stream`	The binary data that is associated with the resource. If the resource has no associated data, this value is null.	Yes

4.8.11.4 REST Protocol

Table 4.149 REST Protocol URI for Download API

Data Type	Description
URL	`https://<server>:8085/ResourceManagement/rest/resources /<resource type>s/<id>/content?contentType=<contentType>`
HTTP Method	`GET`

Data Type	Description
Input/ Output format	XML or JSON
Example	`https://<server>:8085/ResourceManagement/rest/resources/mediafiles/4893/content?contentType=audio/wav`

4.8.11.5 Sequence Diagram

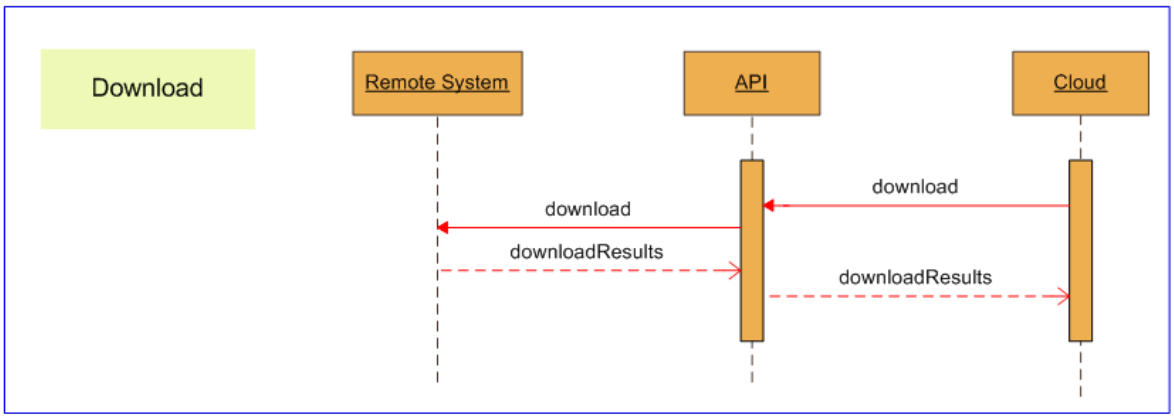


Figure 4.39 Sequence Diagram for Resource Management Download API

4.8.12 Deploy

4.8.12.1 Description

The `Deploy()` API is a generic API that allows bulk update of one or more resources of a single type from a single binary deployment package. The deployment package contains both the information to identify each resource and the data to be associated with that resource. This data may include binary data to be uploaded to a remote system as well as other resource data.

If an associated resource already exists in Unified CCDM, the existing data is replaced with the new data from the deployment package. If the associated resource does not yet exist, it is created.

The type of content being deployed and any additional data required to specify the deployment are determined by the resource type.

The `Deploy()` API encodes the additional data and binary deployment package data in separate parts of a **multipart/form-data** MIME message (see http://tools.ietf.org/html/rfc2388, link checked November 2013).

4.8.12.2 Parameters

The `Deploy()` API expects the following parameters:

Table 4.150 Parameters for Deploy API

SOAP Element Name	Data Type	Description	Required?
`ResourceType`	`String`	The type of resource or resources being deployed.	Yes
`ContentType`	`String`	The type of the deployment package. Depends on the resource type (see Table 4.151).	Yes
`AdditionalData`	`AdditionalData`	Additional data required for the deployment. Depends on the resource type (see section Table 4.151 "Supported Resource Types for Deploy API"). This parameter is encoded in Part 1 of a MIME message.	Yes
`Contents`	`Stream`	The deployment package. This parameter is encoded in Part 2 of a MIME message.	Yes

The supported resource types, and the associated content types and additional data required are:

Table 4.151 Supported Resource Types for Deploy API

Resource Type	Content Type	Content Format	Additional Data		
			Name	Value	Required?
IVR Script	application/zip	ZIP file containing one or more VXML applications created by Cisco Call Studio	FolderId	Folder ID where newly-created resources will be saved.	Yes
			ClusterId	Cluster ID of CVP Operations Console associated with the specified VXML application servers.	Yes
			Servers	Comma separated list of URNs of the VXML application servers where the package is to be deployed. If not present, the package will be deployed to all VXML application servers associated with the specified Cluster ID.	No

4.8.12.3 Return Type

The Deploy() API returns the following objects:

Table 4.152 Return Type for Deploy API

Data Type	Description	Required?
RequestResult[]	Array of results of the deployment.	Yes

4.8.12.4 REST Protocol

Table 4.153 REST Protocol URI for Deploy API

Data Type	Description
URL	https://<server>:8085/ResourceManagement/resources/<resource type>s/deploy?contentType=<contentType>
HTTP Method	POST
Input/ Output format	XML or JSON
Example	https://<server>:8085/ResourceManagement/rest/resources/ivr-scripts/deploy?contentType=application/zip See Table 4.154 for an example MIME message for this URL.

Table 4.154 Example MIME Message for REST Protocol Deploy API

Section	Example Content
Header	`Content-Type: multipart/form-data; boundary="----------8d0a38f0cd807bb"` `Content-Length: 24100`

Part 1: Additional Data
Format: XML or JSON
	`----------8d0a38f0cd807bb`
	`Content-Disposition: form-data; name="additionaldata"`
	`Content-Type: application/json; charset=utf-8`
	`{"Fields":[{"Name":"FolderId","Value": "c2a45666-6137-4109-9744-c82ca730429a"}, {"Name":"Servers","Value":"21,25,29"}, {"Name":"ClusterId","Value": "4356b97f-7466-44bd-ac6b-5de89594cd67"}]}`

Part 2: Content
Format: File
	`----------8d0a38f0cd807bb`
	`Content-Disposition: form-data; name="content"; filename="datafile"`
	`Content-Type: application/zip`
	(deployment package follows as binary data, up to message length defined in header section).

4.8.12.5 Sequence Diagram

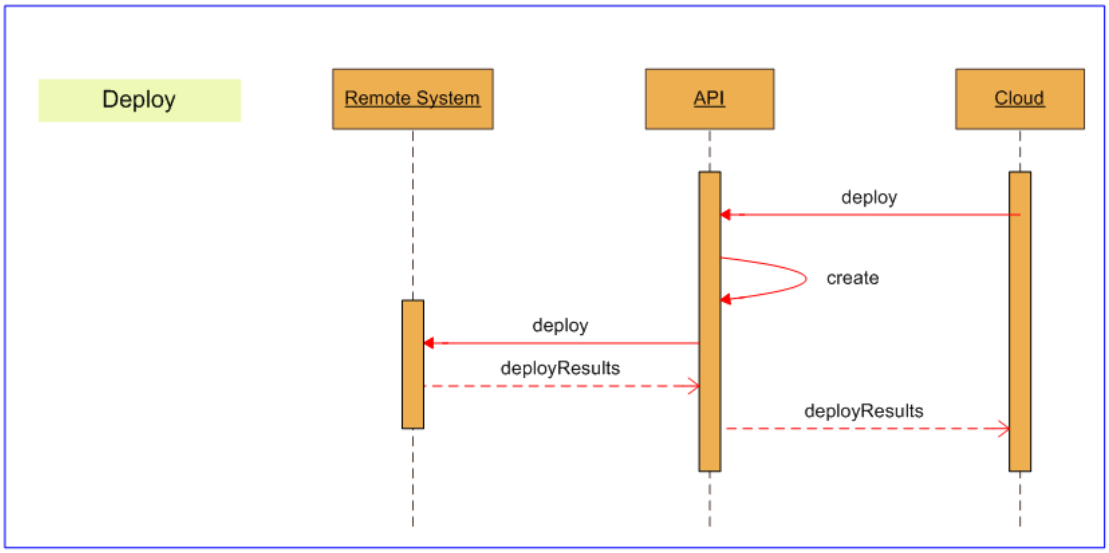


Figure 4.40 Sequence Diagram for Resource Management Deploy API

4.9 Resource Management Web Service Examples

4.9.1 Create Agent

4.9.1.1 Use Case Sequence

Prerequisites:

- The Create Tenant and Create Person use case sequences have been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the peripheral or peripherals that the tenant administrator account allows.

Table 4.155 Actions for Create Agent Use Case

Action	API Calls and Parameters
Retrieve the person record.	`retrieve Id = person URN`
Select the folder in which the agent will be placed. This is typically either the tenant folder or a sub folder underneath the tenant folder.	`search type:folder latest:1 deleted:0`
Find all peripherals that support agents. By default these are peripherals with a client type of 30.	`search type:Peripheral property:ClientType=30 latest:1 deleted:0`
Create the agent linked to the person and peripheral and located in the selected folder. The request will be validated and then queued returning the new Agent URN.	`create Agent`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Agent URN`

4.9.1.2 SOAP Example

The following SOAP request creates a new agent called **Agent1**.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Create>
         <res:resources>
```

```
<!--Zero or more repetitions:-->
<res:Resource>
   <res:Type>Agent</res:Type>
   <res:EffectiveFrom>2009-01-01 00:00:00</res:EffectiveFrom>
   <res:EffectiveTo>2079-06-06 00:00:00</res:EffectiveTo>
  <res:Status>S</res:Status>
   <res:Fields>
      <res:NameValuePair>
         <res:Name>FolderId</res:Name>
         <res:Value>9F44B644-7C24-40CD-9DB9-6D1175DD7FE3</res:Value>
     </res:NameValuePair>
     <res:NameValuePair>
        <res:Name>PeripheralUrn</res:Name>
        <res:Value>-1</res:Value>
     </res:NameValuePair>
     <res:NameValuePair>
         <res:Name>PersonUrn</res:Name>
         <res:Value>4568</res:Value>
      </res:NameValuePair>
      <res:NameValuePair>
        <res:Name>Name</res:Name>
         <res:Value>Agent1</res:Value>
      </res:NameValuePair>
      <res:NameValuePair>
         <res:Name>InternalName</res:Name>
         <res:Value>CICM.Agent1</res:Value>
      </res:NameValuePair>
      <res:NameValuePair>
         <res:Name>Description</res:Name>
         <res:Value>Description of Agent1</res:Value>
      </res:NameValuePair>
      <res:NameValuePair>
         <res:Name>FirstName</res:Name>
         <res:Value>Jim</res:Value>
      </res:NameValuePair>
      <res:NameValuePair>
         <res:Name>LastName</res:Name>
         <res:Value>Smith</res:Value>
      </res:NameValuePair>
      <res:NameValuePair>
         <res:Name>PeripheralNumber</res:Name>
         <res:Value>132456</res:Value>
      </res:NameValuePair>
```

```
                  <res:NameValuePair>
                     <res:Name>PeripheralName</res:Name>
                     <res:Value>Agent1</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Supervisor</res:Name>
                     <res:Value>True</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>DomainName</res:Name>
                     <res:Value>DOMAIN</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>DomainLoginName</res:Name>
                     <res:Value>DOMAIN\Agent1</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>DomainUserName</res:Name>
                     <res:Value>Agent1</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>DomainUserGuid</res:Name>
                     <res:Value>66965475b1d6a448aafc8fb6deac2b2f
 </res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

4.9.2 Update Agent

4.9.2.1 Use Case Sequence

Prerequisites:

- The Create Agent use case sequence has been executed without any existing agent desktop, skill groups and team memberships.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the agent desktop(s), skill group(s) and agent team(s) that the tenant administrator account allows.
- The Agent and associated resources or members are all in the Ready state.

Table 4.156 Actions for Update Agent Use Case

Action	API Calls and Parameters
Retrieve the agent record	`retrieve Id = Agent URN`
Modify the field(s) of interest.	
Update the agent. The agent resource status will go from Ready to Synchronizing until provisioned.	`update Agent`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Agent URN`

The agent re-skilling sequence where this agent is added as a member to a skill group is described in section 4.9.32 "Agent Re-Skilling".

4.9.2.2 SOAP Example

The following SOAP request will update the agent with identity 1234, setting their first name and last name to the passed in values.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Update>
         <res:resources>
            <res:Resource>
               <res:Identity>1234</res:Identity>
<res:Type>Agent</res:Type>
               <res:EffectiveFrom>2009-01-01 00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06 00:00:00</res:EffectiveTo>
               <res:Status>S</res:Status>
               <res:Fields>
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>9F44B644-7C24-40CD-9DB9-6D1175DD7FE3</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                   <res:Name>FirstName</res:Name>
                     <res:Value>Jim</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
```

```
                  <res:Name>LastName</res:Name>
                   <res:Value>Smith</res:Value>
                 </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Update>
   </soap:Body>
</soap:Envelope>
```

4.9.3 Delete Agent

4.9.3.1 SOAP Example

An example of a SOAP request to the Delete API would be as follows: This request will delete the 2 agents with identities 17585 and 17586.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Delete>
         <res:resourceKeys>
            <res:ResourceKey>
               <res:Identity>17585</res:Identity>
               <res:ResourceType>Agent</res:ResourceType>
            </res:ResourceKey>
         <res:ResourceKey>
               <res:Identity>17586</res:Identity>
               <res:ResourceType>Agent</res:ResourceType>
            </res:ResourceKey>
         </res:resourceKeys>
      </res:Delete>
   </soap:Body>
</soap:Envelope>
```

4.9.4 Retrieve Agent

4.9.4.1 SOAP Example

An example of a SOAP request to the Retrieve API to return two agent items from `VW_DIM_AGENT` with the related `AgentUrns` of 17585 and 17586.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
```

```
   <soap:Body>
      <res:Retrieve>
         <!--Optional:-->
         <res:resourceKeys>
            <!--Zero or more repetitions:-->
            <res:ResourceKey>
               <res:Identity>17585</res:Identity>
               <res:ResourceType>Agent</res:ResourceType>
            </res:ResourceKey>
           <res:ResourceKey>
               <res:Identity>17586</res:Identity>
               <res:ResourceType>Agent</res:ResourceType>
            </res:ResourceKey>
         </res:resourceKeys>
      </res:Retrieve>
   </soap:Body>
</soap:Envelope>
```

4.9.5 Search For Agent

4.9.5.1 SOAP Example

An example of a SOAP request calling the Search API to return the results all of the agents in the folder `/Customer1` apart from the one with the ID 17585.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
 <res:Search>
         <!--Optional:-->
         <res:queryString>type:Agent folder:/Customer1</res:queryString>
         <!--Optional:-->
 <res:excludeFilter>
            <!--Zero or more repetitions:-->
            <res:ResourceKey>
               <res:Identity>17585</res:Identity>
               <res:ResourceType>Agent</res:ResourceType>
            </res:ResourceKey>
         </res:excludeFilter>
      </res:Search>
   </soap:Body>
</soap:Envelope>
```

4.9.6 Describe Agent and Person

4.9.6.1 SOAP Example

An example of a SOAP request calling the Describe API to return the metadata description of the agent and person.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
 <res:Describe>
         <!--Optional:-->
         <res:resourceType>
            <!--Zero or more repetitions:-->
            <res:string>Agent</res:string>
         </res:resourceType>
         <res:resourceType>
            <!--Zero or more repetitions:-->
            <res:string>Person</res:string>
         </res:resourceType>
      </res:Describe>
   </soap:Body>
</soap:Envelope>
```

4.9.7 Create Agent Team

4.9.7.1 Use Case Sequence

Prerequisites:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the peripheral(s) that the tenant administrator account allows.

Table 4.157 Actions for Create Agent Team Use Case

Action	API Calls and Parameters
Select the folder in which the agent team will be placed. This is typically either the tenant folder or a sub folder underneath the tenant folder.	`search type:folder latest:1 deleted:0`
Find all peripherals that support agents and agent teams. By default these are peripherals with a client type of 30.	`search type:Peripheral property:ClientType=30 latest:1 deleted:0`
Create the agent team linked to the peripheral and located in the selected folder. The request will be validated and then queued returning its new call type URN.	`create Agent Team`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Agent Team`

4.9.7.2 SOAP Example

The following SOAP request creates an agent team called “testTeam”.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>Agent Team</res:Type>
               <res:EffectiveFrom>1900-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00.0000000</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <!--Zero or more repetitions:-->
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>221C6722-B830-4848-9521-35B2DD8757D7</res:Value>
                  </res:NameValuePair>
```

```
                  <res:NameValuePair>
                     <res:Name>Name</res:Name>
                     <res:Value>testTeam</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>PeripheralUrn</res:Name>
                     <res:Value>-1</res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

4.9.8 Update Agent Team

4.9.8.1 Use Case Sequence

Prerequisites:

- The Create Agent Team use case sequence has been executed.
- The Agent Team and associated resources or members are all in the Ready state.

Table 4.158 Actions for Update Agent Team Use Case

Action	API Calls and Parameters
Retrieve the agent team record	`retrieve Id = Agent Team URN`
Modify the field(s) of interest.	
Update the agent team. The agent team resource status will go from Ready to Synchronizing until provisioned.	`update Agent Team`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Agent Team URN`

4.9.9 Add/Remove Agent Team Members

4.9.9.1 Use Case Sequence

Prerequisites:

- The Create Agent Team use case sequence has been executed.
- The Agent Team and associated resources or members are all in the Ready state.

Table 4.159 Actions for Add/Remove Agent Team Members Use Case

Action	API Calls and Parameters
Retrieve the agent team record. This will have an associated peripheral parent.	`retrieve Id = Agent Team URN`
Find the agents that are already associated with this Agent Team.	`search childof:"Agent Team",<agent team urn>[Status=R;Deleted=0] type:Agent latest:1`
Find all the agents for this peripheral that are not already part of the agent team.	`Search childof:"Peripheral",<peripheral urn>[Status=R;Deleted=0] type:Agent [excluding resources from step2]`
Remove unwanted agents from the agent team. The agent team resource status will go from Ready to Synchronizing until provisioned.	`delete Agent Agent Team Member X N`

Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>`
	or
	`retrieve Id = Agent Team URN`

Add the required agents to the agent team. The agent team resource status will go from Ready to Synchronizing until provisioned.	`Add Agent Agent Team Member X N`

Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>`
	or
	`retrieve Id = Agent Team URN`

4.9.10 Create Call Type

4.9.10.1 Use Case Sequence

Prerequisites:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the peripheral(s) that the tenant administrator account allows.

Table 4.160 Actions for Create Call Type Use Case

Action	API Calls and Parameters
Select the folder in which the call type will be placed. This is typically either the tenant folder or a sub folder underneath the tenant folder.	`search type:folder latest:1 deleted:0`
Create the call type linked to the media routing domain and peripheral and located in the selected folder. The request will be validated and then queued returning its new call type URN.	`create Call Type`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Call Type URN`

4.9.10.2 SOAP Example

The following SOAP request creates a call type called “testCallType”.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>Call Type</res:Type>
               <res:EffectiveFrom>1900-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00.0000000</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <!--Zero or more repetitions:-->
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>221C6722-B830-4848-9521-35B2DD8757D7</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Name</res:Name>
                     <res:Value>testCallType</res:Value>
                  </res:NameValuePair>
               </res:Fields>
```

```
            </res:Resource>
          </res:resources>
        </res:Create>
    </soap:Body>
</soap:Envelope>
```

4.9.11 Update Call Type

4.9.11.1 Use Case Sequence

Prerequisites:

- The Create Call Type use case sequence has been executed without any existing agent memberships.
- The Call Type and associated resources or members are all in the Ready state.

Table 4.161 Actions for Update Call Type Use Case

Action	API Calls and Parameters
Retrieve the Call Type record.	`retrieve Id = Call Type URN`
Modify the field(s) of interest.	
Update the call type. The call type resource status will go from Ready to Synchronizing until provisioned.	`update Call Type`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Call Type URN`

4.9.12 Add/Remove Routing Script Members

4.9.12.1 Use Case Sequence

Prerequisites:

- The Create Agent Team use case sequence has been executed.
- The Agent Team and associated resources or members are all in the Ready state.

Table 4.162 Actions for Add/Remove Routing Script Members Use Case

Action	API Calls and Parameters
Retrieve the Call Type record.	`retrieve Id = Call Type URN`
Find the routing scripts that are already associated with this Call Type.	`memberbychild:"Call Type",37333,"Call Type Routing Script Member"`
Find all the routing scripts that are not already linked to this call type.	`Search type:"Routing Script" status=R Deleted=0 latest:1 [excluding resources from step2]`
Remove unwanted routing scripts from the call type. The call type resource status will go from Ready to Synchronizing until provisioned.	`delete Call Type Routing Script Member X N`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Call Type URN`
Add the required routing scripts from step 3 to the agent team. The agent team resource status will go from Ready to Synchronizing until provisioned.	`Add Call Type Routing Script Member X N`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Call Type URN`

4.9.13 Create Dialed Number

4.9.13.1 Use Case Sequence

Prerequisites:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the routing client(s) and media routing domain (s) that the tenant administrator account allows.

Table 4.163 Actions for Create Dialed Number Use Case

Action	API Calls and Parameters
Select the folder in which the dialed number will be placed. This is typically either the tenant folder or a sub folder underneath the tenant folder.	`search type:folder latest:1 deleted:0`
Find all relevant routing clients. By default these are peripherals with a client type of 13 (IVR) or 30 (PBX).	`search type:"Routing Client" property:ClientType=13 latest:1 deleted:0`
Find the relevant media routing domain; typically this is the default voice domain.	`Search type:"Media Routing Domain" cstext:"Cisco_Voice" latest:1 deleted:0`
Create the dialed number linked to the media routing domain and routing client and located in the selected folder. The request will be validated and then queued returning its new Dialed Number URN.	`create Dialed Number`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Dialed Number URN`

4.9.13.2 SOAP Example

The following SOAP request creates a dialed number called **testDialedNumber**.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Create>
         <res:resources>
            <res:Resource>
         <res:Identity>0</res:Identity>
                <res:Type>Dialed Number</res:Type>
               <res:EffectiveFrom>2010-03-11T12:19:37</res:EffectiveFrom>
               <res:EffectiveTo>2012-03-12T16:32:45.96</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>240253c4-3f30-48c5-a3e6-65b3d94b100e</res:Value>
```

```
                </res:NameValuePair>
                <res:NameValuePair>
                   <res:Name>Name</res:Name>
                   <res:Value>testDialedNumber</res:Value>
                </res:NameValuePair>
                <res:NameValuePair>
                   <res:Name>InternalName</res:Name>
                   <res:Value>DN121</res:Value>
                </res:NameValuePair>
                <res:NameValuePair>
                   <res:Name>Description</res:Name>
                   <res:Value/>
                </res:NameValuePair>
                <res:NameValuePair>
                   <res:Name>RoutingClientUrn</res:Name>
                   <res:Value>22049</res:Value>
                </res:NameValuePair>
                <res:NameValuePair>
                   <res:Name>MediaRoutingDomainUrn</res:Name>
                   <res:Value>10793</res:Value>
                </res:NameValuePair>
                <res:NameValuePair>
                   <res:Name>Digits</res:Name>
                   <res:Value>437457645</res:Value>
                </res:NameValuePair>
                <res:NameValuePair>
                   <res:Name>PermitApplicationRouting</res:Name>
                   <res:Value>False</res:Value>
                </res:NameValuePair>
             </res:Fields>
          </res:Resource>
       </res:resources>
    </res:Create>
 </soap:Body>
</soap:Envelope>
```

4.9.14 Update Dialed Number

4.9.14.1 Use Case Sequence

Prerequisites:

- The Create Dialed Number use case sequence has been executed without any existing agent memberships.

- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the agents that the tenant administrator account allows.
- The Dialed Number and associated resources and members are all in the Ready state.

Table 4.164 Actions for Update Dialed Number Use Case

Action	API Calls and Parameters
Retrieve the Dialed Number record.	`retrieve Id = Dialed Number URN`
Modify the field(s) of interest.	
Update the Dialed Number. The Dialed Number resource status will go from Ready to Synchronizing until provisioned.	`update Dialed number`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Dialed Number URN`

4.9.15 Create Folder

4.9.15.1 SOAP Example

The following SOAP request creates a folder called testfolder2 located under the folder with the ID 00000000-0000-0000-0000-000000000005 (Root).

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
  <soap:Body>
    <res:Create>
      <res:resources>
        <res:Resource>
          <res:Identity>-1</res:Identity>
          <res:Type>Folder</res:Type>
          <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
          <res:EffectiveTo>2079-06-06T00:00:00.0000000</res:EffectiveTo>
          <res:Status>R</res:Status>
          <res:Changestamp>0</res:Changestamp>
          <res:Fields>
            <res:NameValuePair>
              <res:Name>FolderId</res:Name>
```

```
            <res:Value>00000000-0000-0000-0000-000000000005</res:Value>
          </res:NameValuePair>
          <res:NameValuePair>
            <res:Name>Name</res:Name>
            <res:Value>test folder2</res:Value>
          </res:NameValuePair>
          <res:NameValuePair>
            <res:Name>Description</res:Name>
            <res:Value>test description</res:Value>
          </res:NameValuePair>
          <res:NameValuePair>
            <res:Name>InheritPermissions</res:Name>
            <res:Value>true</res:Value>
          </res:NameValuePair>
        </res:Fields>
      </res:Resource>
    </res:resources>
  </res:Create>
 </soap:Body>
</soap:Envelope>
```

4.9.16 Update Folder

4.9.16.1 SOAP Example

The update API may be used for the movement of items between different folders within the Unified CCDM database. Resource updates and move operations may not be performed within the same Update operation and should be achieved using 2 separate web service calls.

To move an item to a new folder the required fields of the resource should be passed in with new field called "NewFolderId". The value of the NewFolderId field should be set to the Unified CCDM identity for the folder to which the resource should be moved.

The following SOAP request moves a Dialed Number resource to another folder:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <Update>
         <resources>
           <Resource>
              <Identity>23221</Identity>
              <Type>Dialed Number</Type>
              <EffectiveFrom>2010-03-15T13:27:12.09</EffectiveFrom>
```

```
                <EffectiveTo>2079-06-06T00:00:00</EffectiveTo>
                <Status>R</Status>
                <Changestamp>0</Changestamp>
                <Fields>
                   <NameValuePair>
                      <Name>NewFolderId</Name>
                      <Value>ec7565cc-0873-43fb-8d6e-31e04b252a27</Value>
                   </NameValuePair>
    <NameValuePair>
                      <Name>FolderId</Name>
                      <Value>1E731B34-D062-404C-B650-B8C45C3BE669</Value>
                   </NameValuePair>
                   <NameValuePair>
                      <Name>Name</Name>
                      <Value>V723.FG.DN</Value>
                   </NameValuePair>
                   <NameValuePair>
                      <Name>RoutingClientUrn</Name>
                      <Value>17801</Value>
                   </NameValuePair>
                   <NameValuePair>
                      <Name>MediaRoutingDomainUrn</Name>
                      <Value>8525</Value>
                   </NameValuePair>
                   <NameValuePair>
                      <Name>Digits</Name>
                      <Value>456</Value>
                   </NameValuePair>
                </Fields>
             </Resource>
          </resources>
       </Update>
   </soap:Body>
</soap:Envelope>
```

The response from the request above is as follows:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/UpdateResponse</a:Action>
      <a:RelatesTo>uuid:713f6333-1589-4b98-a53c-7ed547cfcede</a:RelatesTo>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
```

```
         <u:Timestamp u:Id="_0">
            <u:Created>2010-03-16T12:46:14.013Z</u:Created>
            <u:Expires>2010-03-16T12:51:14.013Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
      <UpdateResponse
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
         <UpdateResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <RequestResult>
               <Identity>23225</Identity>
               <Status>R</Status>
               <Errors i:nil="true"/>
            </RequestResult>
         </UpdateResult>
      </UpdateResponse>
   </s:Body>
</s:Envelope>
```

The movement of this Dialed Number has caused a delete and create operation to be performed. This may happen if an item is moved to a folder outside of its current tenant or if an un-assigned item is moved to a new tenant. This behavior may be recognised by the identity contained within the response not matching that of the original request. The new identity should now be stored and used for all future web service changes relating to this item.

4.9.17 Create Group

4.9.17.1 Use Case Sequence

Prerequisites:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as a either the hoster or the tenant administrator.

Table 4.165 Actions for Create Group Use Case

Action	API Calls and Parameters
Select the folder in which the group will be placed. This is typically either the tenant folder or a sub folder underneath the tenant folder.	`search type:folder latest:1 deleted:0`
Optionally find the security groups to which the new group will be added.	`search type:Group enabled:1 folder:/<tenant>**`
Create the Group and optionally the Group Group members. The request will be executed synchronously returning its Group URN.	`create Group Group Group Member x N`

4.9.17.2 SOAP Example

The following SOAP request creates a group called `testGroup`.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
  <soap:Body>
    <res:Create>
      <res:resources>
        <res:Resource>
          <res:Identity>-1</res:Identity>
          <res:Type>Group</res:Type>
          <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
          <res:EffectiveTo>2079-06-06T00:00:00.0000000</res:EffectiveTo>
          <res:Status>R</res:Status>
          <res:Changestamp>0</res:Changestamp>
          <res:Fields>
            <res:NameValuePair>
              <res:Name>FolderId</res:Name>
              <res:Value>00000000-0000-0000-0000-000000000005</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>Name</res:Name>
              <res:Value>testGroup</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>Description</res:Name>
              <res:Value>test description</res:Value>
            </res:NameValuePair>
          </res:Fields>
```

```
        </res:Resource>
      </res:resources>
    </res:Create>
  </soap:Body>
</soap:Envelope>
```

4.9.18 Update Group

4.9.18.1 Use Case Sequence

Prerequisites:

- The Create Group use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the groups that the tenant administrator account allows.

Table 4.166 Actions for Update Group Use Case

Action	API Calls and Parameters
Retrieve the Group record.	`retrieve Id = User ID`
Modify the field(s) of interest.	
Update the Group.	`update Group`

4.9.19 Create Person

4.9.19.1 Use Case Sequence

Prerequisites:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as either the host administrator or the tenant administrator.

Table 4.167 Actions for Create Person Use Case

Action	API Calls and Parameters
Select the folder in which the person will be placed. This is typically either the tenant folder or a sub folder underneath the tenant folder.	`search type:folder latest:1 deleted:0`
Create the Person in the required folder location. The request will be validated and then queued returning the new Person URN.	`create Person`

Action	API Calls and Parameters
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Person`

4.9.19.2 SOAP Example

The following SOAP request creates a Person record for a person called John Smith.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
     <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>Person</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <!--Zero or more repetitions:-->
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>221C6722-B830-4848-9521-35B2DD8757D7</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Name</res:Name>
                     <res:Value>JohnSmith</res:Value>
                  </res:NameValuePair>
              <res:NameValuePair>
                     <res:Name>FirstName</res:Name>
                     <res:Value>John</res:Value>
                  </res:NameValuePair>
```

```
                  <res:NameValuePair>
                     <res:Name>LastName</res:Name>
                     <res:Value>Smith</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Description</res:Name>
                     <res:Value>A description</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>LoginName</res:Name>
                     <res:Value>john.smith</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>PassPhrase</res:Name>
                     <res:Value>Pa55word</res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

4.9.20 Update Person

4.9.20.1 Use Case Sequence

Prerequisites:

- The Create Person use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the groups that the tenant administrator account allows.

Table 4.168 Actions for Update Person Use Case

Action	API Calls and Parameters
Retrieve the Person record.	`retrieve Id = Person ID`
Modify the field(s) of interest.	
Update the Person.	`update Person`

4.9.21 Create Skill Group

4.9.21.1 Use Case Sequence

Prerequisites:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the peripheral(s) that the tenant administrator account allows.

Table 4.169 Actions for Create Skill Group Use Case

Action	API Calls and Parameters
Select the folder in which the skill group will be placed. This is typically either the tenant folder or a sub folder underneath the tenant folder.	`search type:folder latest:1 deleted:0`
Find all peripherals that support skill groups. By default these are peripherals with a client type of 30.	`search type:Peripheral property:ClientType=30 latest:1 deleted:0`
Find the relevant media routing domain; typically this is the default voice domain	`Search type:"Media Routing Domain" cstext:"Cisco_ Voice" latest:1 deleted:0`
Create the skill group linked to the media routing domain and peripheral and located in the selected folder. The request will be validated and then queued returning its new Skill Group URN.	`create Skillgroup`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Skillgroup URN`

4.9.21.2 SOAP Example

The following SOAP request creates a skill group called **testSkill**.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
```

```
                <res:Identity>-1</res:Identity>
                <res:Type>SkillGroup</res:Type>
                <res:EffectiveFrom>1900-01-01T00:00:00</res:EffectiveFrom>
                <res:EffectiveTo>2079-06-06T00:00:00.0000000</res:EffectiveTo>
                <res:Status>R</res:Status>
                <res:Changestamp>0</res:Changestamp>
                <res:Fields>
                   <!--Zero or more repetitions:-->
                   <res:NameValuePair>
                      <res:Name>FolderId</res:Name>
                      <res:Value>221C6722-B830-4848-9521-35B2DD8757D7</res:Value>
                   </res:NameValuePair>
                   <res:NameValuePair>
                      <res:Name>Name</res:Name>
                      <res:Value>testSkill</res:Value>
                   </res:NameValuePair>
                   <res:NameValuePair>
                      <res:Name>PeripheralUrn</res:Name>
                      <res:Value>-1</res:Value>
                   </res:NameValuePair>
                   <res:NameValuePair>
                      <res:Name>MediaRoutingDomainUrn</res:Name>
                      <res:Value>8345</res:Value>
                   </res:NameValuePair>
                   <res:NameValuePair>
                      <res:Name>PeripheralName</res:Name>
                      <res:Value>testSkill</res:Value>
                   </res:NameValuePair>
                   <res:NameValuePair>
                      <res:Name>AvailableHoldoffDelay</res:Name>
                      <res:Value>0</res:Value>
                   </res:NameValuePair>
                   <res:NameValuePair>
                      <res:Name>Priority</res:Name>
                      <res:Value>0</res:Value>
                   </res:NameValuePair>
                </res:Fields>
             </res:Resource>
          </res:resources>
       </res:Create>
    </soap:Body>
 </soap:Envelope>
```

4.9.22 Update Skill Group

4.9.22.1 Use Case Sequence

Prerequisites:

- The Create Skill Group use case sequence has been executed without any existing agent memberships.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the agents that the tenant administrator account allows.
- The Skill group and associated resources or members are all in the Ready state.

Table 4.170 Actions for Update Skill Group Use Case

Action	API Calls and Parameters
Retrieve the skill group record	`retrieve Id = Skillgroup URN`
Modify the field(s) of interest.	
Update the skill group. The skill group resource status will go from Ready to Synchronizing until provisioned.	`update SkillGroup`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Skillgroup URN`

The agent re-skilling sequence where this agent is added as a member to a skill group is described in section 4.9.27 "Agent to Skill Group Membership".

4.9.23 Create Tenant

4.9.23.1 Use Case Sequence

Prerequisites:

- The caller of the Unified CCDM Web Services is logged in as the host administrator.
- The underling equipment is located in the folder structure.

Table 4.171 Actions for Create Tenant Use Case

Action	API Calls and Parameters

Select the equipment clusters that this tenant will be placed in.	`search type:"Cluster Resource" property:"ResourceTypeInternalName" =CRT_CICM`
	and/or
	`search type:"Cluster Resource" property:"ResourceTypeInternalName" =CRT_CCM`

Create the tenant, setting the `MappedClusterResources` field to the comma separated cluster resources found in step 1. The request will be validated and then queued, returning its new Tenant URN.	`create Tenant`

Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<TenantUrn>`
	or
	`retrieve Id=TenantUrn`

Create one or more tenant users under the new tenant (see Create User use case sequence in section 4.9.25 "Create User"). This would typically be one or more Tenant administrator accounts. Tip. The `FolderId` field for the User can be determined by retrieving the Tenant record in step 3 or by supplying the path text which will be "/" + tenant name set in step 2.	`create User`

4.9.24 Update Tenant

4.9.24.1 Use Case Sequence

Prerequisites:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as either the host administrator or the tenant administrator.
- The Tenant and associated resources or members are all in the Ready state.

Table 4.172 Actions for Update Tenant Use Case

Action	API Calls and Parameters
Retrieve the Tenant record.	`retrieve Id = Tenant URN`
Modify the field(s) of interest.	
Update the Tenant. The Tenant resource status will go from Ready to Synchronizing until provisioned.	`update Tenant`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Tenant URN`

4.9.25 Create User

4.9.25.1 Use Case Sequence

Prerequisites:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as a host or tenant administrator.

Table 4.173 Actions for Create User Use Case

Action	API Calls and Parameters
Select the folder in which the user will be placed. This is typically either the tenant folder or a sub folder underneath the tenant folder.	`search type:folder latest:1 deleted:0`
Optionally find the security groups to which the new user will be added.	`search type:Group enabled:1 folder:/<tenant>**`
Create the User and optionally the User Group members. The request will be executed synchronously returning its User URN.	`create User, User Group Member xN`

4.9.25.2 SOAP Example

The following SOAP request creates a user called **testUser**.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
  <soap:Body>
    <res:Create>
      <res:resources>
        <res:Resource>
          <res:Identity>-1</res:Identity>
          <res:Type>User</res:Type>
          <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
          <res:EffectiveTo>2079-06-06T00:00:00.0000000</res:EffectiveTo>
          <res:Status>R</res:Status>
          <res:Changestamp>0</res:Changestamp>
          <res:Fields>
            <res:NameValuePair>
              <res:Name>FolderId</res:Name>
              <res:Value>00000000-0000-0000-0000-000000000005</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>Description</res:Name>
              <res:Value>test description</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>LoginName</res:Name>
             <res:Value>testuser</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>PassPhrase</res:Name>
              <res:Value>Pa55word</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>FirstName</res:Name>
              <res:Value>First</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>LastName</res:Name>
              <res:Value>Last</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>Email</res:Name>
              <res:Value>email@exony.com</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>PassPhraseNeverExpires</res:Name>
              <res:Value>false</res:Value>
```

```
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>PassPhraseChangeEnabled</res:Name>
              <res:Value>true</res:Value>
            </res:NameValuePair>
            <res:NameValuePair>
              <res:Name>ForceChangePassword</res:Name>
              <res:Value>false</res:Value>
            </res:NameValuePair>
          </res:Fields>
        </res:Resource>
      </res:resources>
    </res:Create>
  </soap:Body>
</soap:Envelope>
```

Time Zones

Users can be created using the default server time-zone information (as in the request above), or they can optionally be created with the time-zone specified at creation time. To specify the time zone in the create request, add the optional parameter `TimeZone` and supply the name of the time zone.

The list of valid timezones is contained in the database table `TE_ADM_TIME_ZONE` and is reproduced in Appendix C "Valid Time Zones".

4.9.26 Update User

4.9.26.1 Use Case Sequence

Prerequisites:

- The Create User use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the groups that the tenant administrator account allows.

Table 4.174 Actions for Update User Use Case

Action	API Calls and Parameters
Retrieve the User record.	`retrieve Id = User ID`
Modify the field(s) of interest.	
Update the User	`update User`

4.9.27 Agent to Skill Group Membership

4.9.27.1 SOAP Example

The following SOAP request creates a membership between an agent and a skill group where the identity of the agent is **7565** and the identity of the skill group is **17701**.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>Agent SkillGroup Member</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00.0000000</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <!--Zero or more repetitions:-->
                  <res:NameValuePair>
                     <res:Name>ParentId</res:Name>
                     <res:Value>17701</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>ChildId</res:Name>
                     <res:Value>7565</res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

4.9.28 Agent to Agent Team Membership

4.9.28.1 SOAP Example

The following SOAP request creates a membership between an agent and an agent team where the identity of the agent is **7565** and the identity of the agent team is **7917**.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>Agent Agent Team Member</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <!--Zero or more repetitions:-->
                  <res:NameValuePair>
                     <res:Name>ParentId</res:Name>
                     <res:Value>7917</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>ChildId</res:Name>
                     <res:Value>7565</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Supervisor</res:Name>
                     <res:Value>false</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>PrimarySupervisor</res:Name>
                     <res:Value>false</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>PhysicalMember</res:Name>
                     <res:Value>true</res:Value>
                  </res:NameValuePair>
```

```
            </res:Fields>
         </res:Resource>
      </res:resources>
   </res:Create>
</soap:Body>
</soap:Envelope>
```

4.9.29 User to Group Membership

4.9.29.1 SOAP Example

The following SOAP request adds the user with the identity **80479952-4236-456d-9dc4-668a14a9f8a9** to the group with the identity **7d80aa2f-d649-40d8-9c79-0036352d64d2**.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>User Group Member</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00.0000000</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <!--Zero or more repetitions:-->
                  <res:NameValuePair>
                     <res:Name>ParentId</res:Name>
                     <res:Value>7d80aa2f-d649-40d8-9c79-0036352d64d2</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>ChildId</res:Name>
                     <res:Value>80479952-4236-456d-9dc4-668a14a9f8a9</res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
```

```
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

4.9.30 Group to Group Membership

4.9.30.1 SOAP Example

The following SOAP request adds the group with the identity **7d80aa2f-d649-40d8-9c79-0036352d64d2** to the group with the identity **bf272315-0435-49bd-a117-d2117754593a**.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>Group Group Member</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00.0000000</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <!--Zero or more repetitions:-->
                  <res:NameValuePair>
                     <res:Name>ParentId</res:Name>
                     <res:Value>bf272315-0435-49bd-a117-d2117754593a</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>ChildId</res:Name>
                     <res:Value>7d80aa2f-d649-40d8-9c79-0036352d64d2</res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

4.9.31 Dialed Number to Call Type Membership

4.9.31.1 SOAP Example

The following SOAP request associates the Dialed Number with the identity **9149** to the Call Type with the identity **8909**. Note that the Dialed Number Call Type Membership has a number of properties that may be passed in during create and edit operations.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Create>
         <res:resources>
            <res:Resource>
               <res:Identity>0</res:Identity>
               <res:Type>Dialed Number Call Type Member</res:Type>
               <res:EffectiveFrom>2010-03-11T12:19:37</res:EffectiveFrom>
               <res:EffectiveTo>2012-03-12T16:32:45.96</res:EffectiveTo>
               <res:Status>S</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <res:NameValuePair>
                     <res:Name>ParentId</res:Name>
                     <res:Value>8909</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>ChildId</res:Name>
                     <res:Value>9149</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Item</res:Name>
                     <res:Value>0</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>CLIWildCardType</res:Name>
                     <res:Value>5</res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

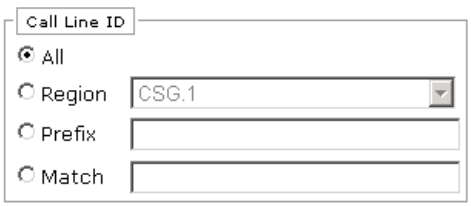
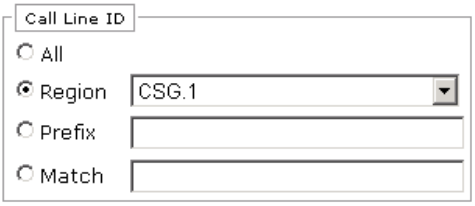
The Dialed Number Call Type membership has a number of fields available on it and may be configured in different ways for Dialed Numbers that belong to Voice or non-Voice Media Routing Domains.

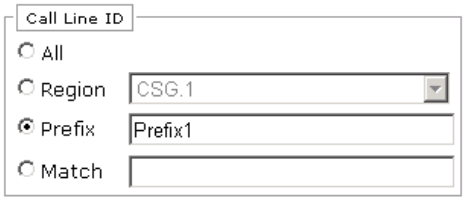
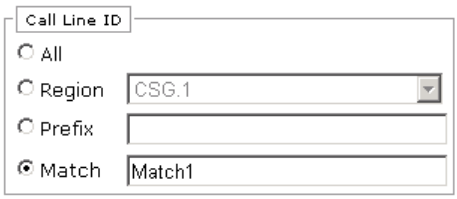
The fields that may be set when creating or updating a Dialed Number Call Type Membership are:

- `ParentId` (required) the identity of the Call Type resource
- `ChildId` (required) the identity of the Dialed Number resource
- `RegionUrn`
- `Item` (required)
- `CLIWildCard`
- `CLIWildCardType` (required)
- `CEDWildCardDescription`

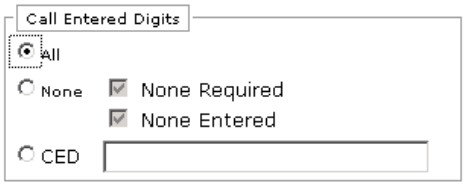
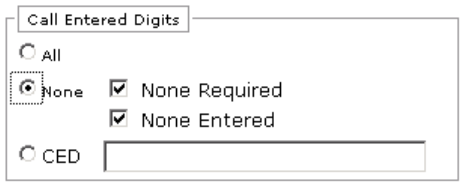
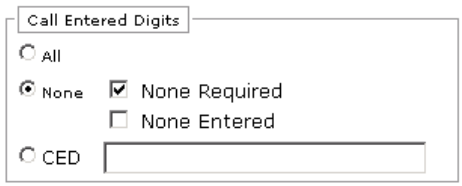
When creating or editing a Dialed Number connected to a Voice Media Routing Domain, the configuration can be specified as follows.

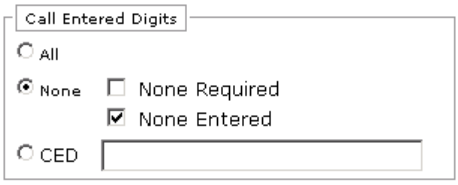
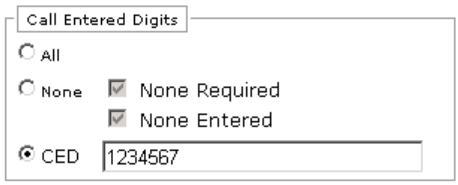
Call Line ID Configuration

	CliWildCardType 5 CliWildCard Null RegionUrn Null
	CliWildCardType 4 CliWildCard Null RegionUrn The Identity of the selected Region

	CliWildCardType 6 CliWildCard Prefix1 RegionUrn Null
	CliWildCardType 3 CliWildCard Match1 RegionUrn Null

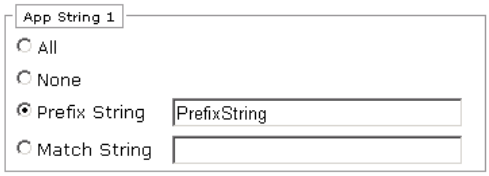
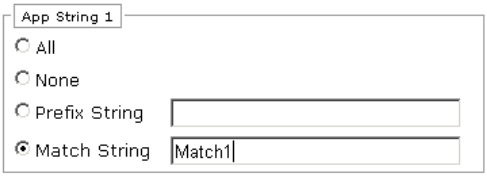
Call Entered Digits Configuration

	CedWildCard _A
	CedWildCard _N
	CedWildCard _NR

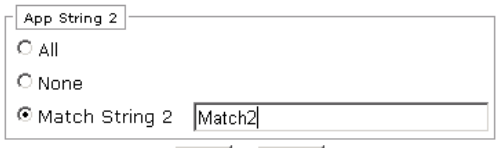
	CedWildCard _NE
	CedWildCard 1234567

App String 1 Configuration

When creating or editing a Dialed Number connected to a non-Voice Media Routing Domain the configuration can be specified as follows.

	CliWildCardType 5 CliWildCard Null RegionUrn Null
	CliWildCardType 3 CliWildCard Null RegionUrn Null
	CliWildCardType 6 CliWildCard PrefixString RegionUrn Null
	CliWildCardType 3 CliWildCard Match1 RegionUrn Null

App String 2 configuration

	CedWildCard _A
	CedWildCard Null
	CedWildCard Match2

4.9.32 Agent Re-Skilling

The following example describes the Resource Management Web Service API usage when performing a re-skilling action from a third party client.

The example is a third party application that first lists Skill Groups that the user Bob may see. Upon selection of one of those Skill Groups a list of Agents that are currently working in that Skill is presented along with a list of Agent available to work in that Skill that Bob can see. Agents may be removed and added to and from the Skill Group and when complete the "Save" button may be pressed to save the changes to the Unified CCDM database and the remote system.

The web service API interaction for this example is as follows:

Step 1

Retrieve the list of Skill Groups that the user can see.

Call to Search API passing in the search string:

```
type:SkillGroup folder:/Tenant latest:1 max:50 status:R
```

Step 2

The user selects the Skill Group that they wish to change (for example, identity 1234) and is presented with the list of Agents already in that Skill Group

Call to the Search API with the search string

```
childof:SkillGroup,1234>[Status=R;Deleted=0] type:Agent latest:1
max:50
```

Step 3

The Peripheral identity of the Peripheral that the Skill Group is on is retrieved

Call to the Search API with the search string:

```
parentof:SkillGroup,1234 type:Peripheral
```

Step 4

The Peripheral identity (4321) is used to get the list of possible Agents that may be added to the Skill Group

Call to Search API with the search string

```
childof:Peripheral,4321 type:Agent
```

The API is also passed a collection of `ResourceKeys` of the selected Agents to exclude them from the list

Step 5

The user uses the client to create and remove Skill Group/Agent members as they require. The user presses the "Save" button to commit the change to Unified CCDM database.

Step 6

The 'to delete existing memberships' and 'to add new memberships' collections built up with the previous search queries are maintained by the client. A call to the Delete API is then made passing the 'to delete collection' of Resource objects with the same type set and the parent/child identities of the relationships to be removed. A subsequent call to the Create API is made, passing the 'to add' collection of Resource objects with the type AgentSkillGroupMember and the parent and child identities set accordingly.

Note

Unified CCDM performs a number of validation and capacity checks when performing provisioning requests. It is recommended when provisioning resource memberships that Delete procedures are performed before create to reduce the risk of capacity exceptions occurring.

4.9.33 User Creation

The following example describes the usage of the Web Service APIs to create a new Unified CCDM user account. This account will be able to perform item create/ edit and delete operations for a specific tenant.

When a new tenant is created in Unified CCDM (for example, when a new Unified CCE Customer Definition is imported from Unified CCE), 3 security groups are automatically configured to allow for simplistic configuration of security for newly added users.

The automatically created groups are:

- Advanced Users (high level access, intended for tenant administrator accounts)
- Basic Users (low level access, intended for reporting only accounts)
- Supervisor Users (medium level access, intended for supervisors who require access to reports and permissions to re-skill agents).

An example showing the web service API interactions to create a tenant admin account is as follows:

Step 1

Retrieve the Folder ID for the for the tenant folder. Call the Search API with the following search string:

```
type:IT_FOLDER folder:/ name:"v723"
```

where `v723` is the name of the tenant. The returning items identity field will be the `FolderID` that will be used later

Step 2

Create a new user in the tenant folder. Call the Create API specifying `User` as the Type, `FolderID` as the folder id retrieved above and `Name/LoginName` as the name of the user that is to be created. An initial password must also be specified. The create API will return the identity of the new user if successful which will be used later

Step 3

Locate the Advanced Users group for the tenant that we have created the new user in. Call the Search API with the following search string:

```
type:IT_GROUP folder:"221c6722-b830-4848-9521-35b2dd8757d7"
name:"Advanced Users"
```

where `221c6722-b830-4848-9521-35b2dd8757d7` is the identity for the tenant's folder retrieved in step 1.

Step 4

Create a new User/Group member between the new user and the Advanced Users group for the users tenant Call the Create API specifying `UserGroupMember` as the Type, the new user id retrieved in step 2 as the `ChildId` and the group id retrieved in step 3 as the `ParentId.`

Step 5

The new user has been created and added to the Advanced Users group.

4.9.34 Move Unallocated Dialed Number to Tenant Folder

An example showing the web service API interactions to move an unallocated dialed number(s) to a pre-created call type/routing script is as follows:

Step 1

Get the destination Tenant folder id. Call the Search API with the following search string:

```
type:IT_FOLDER folder:/ name:"v723"
```

where v723 is the name of the tenant.

The returning items identity field will be the `FolderID` that will be used later.

Step 2

Get the list of unallocated Dialed Numbers that the tenant login can see. Note: these are typically located either in the `/Unallocated/<equipment> folder` (by default on a multi-tenant Unified CCE) or manually moved to a product offering specific folder in `/Shared`, for example, `/Shared/SFDC/Bronze.`

Call to Search API passing in the search string

```
type:"Dialed Number" folder:"<unallocated DNs folder>" max:"<desired
number> offset:0 latest:1 deleted:0
```

For example,

```
type:"Dialed Number" folder:"/Shared/SFDC/Bronze" max:3 offset:0
latest:1 deleted:0
```

Step 3

Move the returned dialed numbers to the required tenant folder. For each dialed number in the returned array set its folder id to the tenant folder id found in step 1. Call Update AI with the modified dialed numbers.

Step 4

The Dialed Numbers have been moved from the unallocated location to the tenant folder.

4.9.35 Link Dialed Number to Call Type/Routing Script

Mapping the call treatment for a Dialed Number is ensuring that the Dialed Number is linked to the correct Call Type which in turn is linked to the correct

Routing Script which contains the call treatment logic. This use case assumes that the Dialed Number has been moved to the correct tenant folder and that the Call Type(s) are already associated with Routing Script(s).

The web service API interactions to link an dialed number in a tenant folder to an existing call type/routing script would be as follows:

Step 1

Get the Dialed Numbers anywhere in the tenant specific folder structure. Call to Search API passing in the search string:

```
type:"Dialed Number" folder:/<tenant location>**" max:<desired number>
offset:0 latest:1 deleted:0 status:R
```

For example,

```
type:"Dialed Number" folder:/Acme**" max:5 offset:0 latest:1 deleted:0
```

Step 2

Get the Call Types anywhere in the tenant specific folder structure.

Call to Search API passing in the search string:

```
type:"Call Typer" folder:/<tenant location>**" max:<desired number>
offset:0 latest:1 deleted:0
```

For example,

```
type:"Call Type" folder:/Acme**" max:5 offset:0 latest:1 deleted:0
```

Step 3

Add the relevant dialed number(s) to the relevant call type(s) as shown in section 4.9.31 "Dialed Number to Call Type Membership".

Call the Create API for resource type "Dialed Number Call Type Member" with parent and child item urns set to the relevant Call Type and Dialed Number urns found in the previous two searches.

Step 4

The new dialed number call type memberships will be added to the Unified CCDM database and their membership URNs returned to the caller for tracking purposes.

Step 5

To check for any existing memberships between call types and dialed numbers then the following Search APIs can be called.

To find if a call type has any associated dialed numbers either

```
childof:"Call Type",<call type urn> type:"Dialed Number" latest:1
```

or

```
memberbyparent:"Call Type",<call type urn>,"Dialed Number Call Type
Member"
```

To find if a dialed number has any associated call types, either

```
parentof:"Dialed Number",<dialed number urn> type:"Call Type" latest:1
```

or

```
memberbychild:"Dialed Number",<dialed number urn>, "Dialed Number Call
Type Member"
```

4.9.36 Unlink Dialed Number to Call Type/Routing Script

This example details the steps required to remove a membership between a Dialed Number and Call Type; for example when the Dialed Number is to be moved to a different Call Type. The example assumes that the Dialed Number is already linked to the Call Type and that both the resources and their membership is in the Ready state (that is, they can be provisioned). The web service API interactions in this case would be as follows:

Step 1

Get the Dialed Numbers anywhere in the tenant specific folder structure.

Call to Search API passing in the search string:

```
type:"Dialed Number" folder:/<tenant location>**" cstext:<dialed
number name> status:R latest:1
```

for example,

```
type:"Dialed Number" folder:/Acme**" cstext:"CCMIST_RC1.2551" status:R
latest:1
```

Step 2

Find its existing Dialed Number Call Type Membership(s).

Call to Search API passing in the search string:

```
memberbychild:"Dialed Number",<dialed number urn>, "Dialed Number Call
Type Member" latest:1 status:R
```

Step 3

Delete the membership(s).

Call the Delete API for resource type "Dialed Number Call Type Member" with resource type "Dialed Number Call Type Member" and Identity set to the identities found in the previous membership search..

Step 4

The existing dialed number call type memberships will be marked for deletion in the Unified CCDM database and will be provisioned in due course. Note that the memberships will be purged from the underlying Unified CCE in this case but will remain in the Unified CCDM database with a status of Deleted for reporting and tracking purposes.

Step 5

Subscribers who have subscribed to this resource type will receive a notification for each step in the state machine cycle until the operation is either successful or fails provisioning.

4.9.37 Rename Resources

This example details the steps required to rename the resources for a specific tenant in the underlying equipment. This would typically be done when activating a template or 'blank' customer in which all the resources are pre-allocated with a template prefix name and is done for operational tracking and having meaningful names in the end user reports. The example assumes that all the resources and their membership are in the Ready state, that is, they can be provisioned. The web service API interactions in this case would be as follows:

Step 1

Get the resources underneath the tenant:

Call to Search API passing in the search string:

```
folder:/<tenant folder>** status:R latest:1 max:50 offset:0
```

This will return the all the resources under the tenant folders. Those that are useful to rename for reporting are Tenant, Call Types and Skill Groups. Routing Scripts, Network Vru Scripts and Routes may be usefully renamed for operation tracking. Several calls may be needed using the max and offset keywords to page the data.

Step 2

Rename each resource returned in the previous (including the tenant resource):

Call the Update API changing both the resources internal name and name fields to the desired tenant template name. Typically each resource type has its own naming convention that is specific to each installation

For example; a pattern for Network VRU Scripts may be `<Tenant Name>.<N>.VRU` and the caller could simply rename the tenant prefix, for example, `Blank103.1.1.VRU` to `Acme.1.1.VRU`. The rename mask should ensure that it matches the rules for resource names in terms of length and legal characters.

4.9.38 Create Agent with Pkey Maps, Method 1 (Explicit Pkey Maps)

4.9.38.1 Description

This example creates an Agent that is linked to two different remote equipment instances. In this example, the pkey map is specified explicitly and separately for each remote equipment instance.

Tip

Use separate pkey maps if an item has different details on different remote equipment instances.

You can mix this method with method 2 (implicit pkey maps). You can specify some fields with the main item as common fields for a list of equipment instances and specify some fields individually for each equipment instance as pkey maps.

4.9.38.2 Use Case Sequence

Prerequisites:

- The Create Tenant and Create Person use case sequences have been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the peripheral or peripherals that the tenant administrator account allows.

Table 4.175 Create Agent with Pkey Maps, Method 1 (Explicit Pkey Maps)

Action	API Calls and Parameters
Create the Agent, specifying the Agent details and pkey maps as in the SOAP example below.	`create Agent`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Agent Id`

4.9.38.3 SOAP Example

The following SOAP request creates an agent, with first name “Jim” and last name “Smith”, that is linked to two agents (“Jim Smith A” and “Jim Smith B”) on two remote equipment instances.

In this example, two pkey maps are specified, one for each equipment instance.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Create>
         <res:resources>
            <res:Resource>
               <res:Type>Agent</res:Type>
               <res:EffectiveFrom>2013-01-01 00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06 00:00:00</res:EffectiveTo>
               <res:Status>S</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>9F44B644-7C24-40CD-9DB9-6D1175DD7FE3</res:Value>
                  </res:NameValuePair>
                 <res:NameValuePair>
                     <res:Name>PeripheralUrn</res:Name>
                    <res:Value>-1</res:Value>
                 </res:NameValuePair>
                 <res:NameValuePair>
                     <res:Name>PersonUrn</res:Name>
                     <res:Value>4568</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>FirstName</res:Name>
                     <res:Value>Jim</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>LastName</res:Name>
                     <res:Value>Smith</res:Value>
                  </res:NameValuePair>
               </res:Fields>
               <res:EquipmentMapping>
                  <res:Equipment>
                     <res:Type>Agent Pkey</Type>
                     <res:Fields>
                        <res:NameValuePair>
                           <res:Name>Name</Name>
                           <res:Value>Jim Smith A</Value>
                        </res:NameValuePair>
```

```
                        <res:NameValuePair>
                           <res:Name>InternalName</Name>
                           <res:Value/>
                        </res:NameValuePair>
                        <res:NameValuePair>
                           <res:Name>Description</Name>
                           <res:Value>Jim Smith on Equipment A</Value>
                        </res:NameValuePair>
                        <res:NameValuePair>
                           <res:Name>ClusterResourceId</Name>
                           <res:Value>{c5ed5103-55cc-420b-8fa9-9d64c2eaa339}
</Value>
                        </res:NameValuePair>
                     </res:Fields>
                  </res:Equipment>
                  <res:Equipment>
                     <res:Type>Agent Pkey</Type>
                     <res:Fields>
                        <res:NameValuePair>
                           <res:Name>Name</Name>
                           <res:Value>Jim Smith B</Value>
                        </res:NameValuePair>
                        <res:NameValuePair>
                           <res:Name>InternalName</Name>
                           <res:Value/>
                        </res:NameValuePair>
                        <res:NameValuePair>
                           <res:Name>Description</Name>
                           <res:Value>Jim Smith on Equipment B</Value>
                        </res:NameValuePair>
                        <res:NameValuePair>
                           <res:Name>DomainLoginName</Name>
                           <res:Value>Jim</Value>
                        </res:NameValuePair>
                        <res:NameValuePair>
                           <res:Name>ClusterResourceId</Name>
                           <res:Value>{5103c5ed-d43e-fa4e-98f0-2eaa33a9d64c}
</Value>
                        </res:NameValuePair>
                     </res:Fields>
                  </res:Equipment>
               </res:EquipmentMapping>
```

```
            </res:Resource>
          </res:resources>
        </res:Update>
    </soap:Body>
</soap:Envelope>
```

4.9.39 Create Agent Team with Pkey Maps, Method 2 (Implicit Pkey Maps)

4.9.39.1 Description

This example creates an Agent Team that is linked to two different remote equipment instances. In this example, the remote equipment instances are specified in a comma-separated list as part of the main Agent Team item. When the Agent Team is created, a pkey map containing the relevant data will be created for each remote equipment instance.

Tip

Use this method if an item has the same details on different remote equipment instances.

You can mix this method with method 1 (explicit pkey maps). You can specify some fields with the main item as common fields for a list of equipment instances and specify some fields individually for each equipment instance as pkey maps.

4.9.39.2 Use Case Sequence

Prerequisites:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCDM Web Services is logged in as the tenant administrator and only has access to the peripheral or peripherals that the tenant administrator account allows.

Table 4.176 Create Agent Team, Two Pkey Maps, Method 2 (Combined Pkey Maps)

Action	API Calls and Parameters
Create the Agent Team, specifying the Agent Team details and pkey maps as in the SOAP example below.	`create AgentTeam`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = AgentTeam Id`

4.9.39.3 SOAP Example

The following SOAP request creates an Agent Team, “Test Agent Team”, that is linked to two Agent Teams on two remote equipment instances, where the two Agent Teams have identical details on both equipment instances.

In this example, the remote equipment instances are specified in a comma-separated list.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Create>
         <res:resources>
            <res:Resource>
               <res:Type>Agent Team</res:Type>
               <res:EffectiveFrom>2013-01-01 00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06 00:00:00</res:EffectiveTo>
               <res:Status>S</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>221C6722-B830-4848-9521-35B2DD8757D7</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Name</res:Name>
                     <res:Value>Test Agent Team</res:Value>
                  </res:NameValuePair>
```

```
                  <res:NameValuePair>
                     <res:Name>PeripheralUrn</res:Name>
                     <res:Value>-1</res:Value>
                  </res:NameValuePair>
                  <NameValuePair>
                     <Name>MappedClusterResources</Name>
                     <Value>{c5ed5103-55cc-420b-8fa9-9d64c2eaa339},
                            {5103c5ed-d43e-fa4e-98f0-2eaa33a9d64c}</Value>
                  </NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Update>
   </soap:Body>
</soap:Envelope>
```

4.9.40 Update Agent (Additional Pkey Map)

4.9.40.1 Description

This example adds an additional remote equipment mapping to an existing Agent.

4.9.40.2 Use Case Sequence

Prerequisites:

- The specified Agent already exists.

Table 4.177 Update Agent (Additional Remote Equipment Mapping) Use Case

Action	API Calls and Parameters
Retrieve the agent record	`retrieve Id = AgentId`
Modify the agent record to include a pkey map containing the mapping to the additional remote equipment.	
Update the agent, specifying the remote equipment mapping in a pkey as in the SOAP example below.	`update Agent`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = AgentId`

4.9.40.3 SOAP Example

The following SOAP request updates the agent with identity 1234 to include a link to an additional remote equipment instance.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Update>
         <res:resources>
            <res:Resource>
               <res:Identity>1234</res:Identity>
               <res:Type>Agent</res:Type>
               <res:Fields>
               </res:Fields>
               <res:EquipmentMapping>
                  <res:Equipment>
                     <res:Type>Agent Pkey</Type>
                     <res:Fields>
                        <res:NameValuePair>
                           <res:Name>Name</Name>
                           <res:Value>Jim Smith X</Value>
                        </res:NameValuePair>
                        <res:NameValuePair>
                           <res:Name>InternalName</Name>
                           <res:Value/>
                        </res:NameValuePair>
                        <res:NameValuePair>
                           <res:Name>Description</Name>
                           <res:Value>Jim Smith on Equipment X</Value>
                        </res:NameValuePair>
                        <res:NameValuePair>
                           < <res:Name>ClusterResourceId</Name>
                           <res:Value>{c5ed5103-55cc-420b-8fa9-9d64c2eaa33b}
</Value>
                       </res:NameValuePair>
                     </res:Fields>
                  </res:Equipment>
               </res:EquipmentMapping>
            </res:Resource>
         </res:resources>
      </res:Update>
   </soap:Body>
</soap:Envelope>
```

4.9.41 Update Agent (Delete a Pkey Map)

4.9.41.1 Description

This example deletes one of the remote equipment mappings for an Agent.

Note
You cannot delete the last remaining remote equipment mapping unless you delete the Agent.

4.9.41.2 Use Case Sequence

Prerequisites:

- The specified Agent already exists and has at least two pkey map items.

Table 4.178 Update Agent (Additional Remote Equipment Mapping) Use Case

Action	API Calls and Parameters
Retrieve the Agent record.	`retrieve Id = Agent Id`
Identify the id of the Agent Pkey to be deleted.	
Delete the Agent Pkey as in the SOAP example below.	`delete Agent Pkey`
If required, wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = Agent Pkey Id`

4.9.41.3 SOAP Example

The following SOAP request deletes the Agent pkey map with id 17585 from Agent 1234. Any other Agent pkey maps associated with Agent 1234 are unaffected.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Update>
         <res:resources>
            <res:Resource>
               <res:Identity>1234</res:Identity>
```

```
            <res:Type>Agent</res:Type>
            <res:Fields>
            </res:Fields>
            <res:EquipmentMapping>
               <res:Equipment>
                  <res:Type>Agent Pkey</Type>
                 <res:Identity>17585</res:Identity>
                 <res:Status>D</res:Status>
                  <res:Fields></res:Fields>
               </res:Equipment>
            </res:EquipmentMapping>
         </res:Resource>
      </res:resources>
   </res:Update>   </soap:Body>
</soap:Envelope>
```

4.9.42 Create Agent To Agent Team Membership (Two Remote Equipment Mappings)

4.9.42.1 Description

This example creates an Agent to Agent Team membership that relates an Agent to an Agent Team where the Agent and the Agent Team are both linked to two different remote equipment instances. In this example, the Agent to Agent Team memberships for both remote equipment instances are specified when the Agent to Agent Team membership is created.

4.9.42.2 Use Case Sequence

Prerequisites:

- The specified Agent exists and is mapped to at least two remote equipment instances.
- The specified Agent Team exists and is mapped to the same two remote equipment instances.

Table 4.179 Create Agent To Agent Team Membership (Two Remote Equipment Mappings) Use Case

Action	API Calls and Parameters
Retrieve the Agent record and obtain the Agent Pkey ids for each remote equipment mapping.	`retrieve Id = Agent Id`
Retrieve the Agent Team record and obtain the Agent Pkey ids for each remote equipment mapping.	`retrieve Id = Agent Team Id`
Create the Agent Agent Team membership, specifying the remote equipment mapping in a pkey as in the SOAP example below.	`update Agent`

4.9.42.3 SOAP Example

The following SOAP request will create a membership between the Agent with id 1234 and pkey maps of 1111 and 2222 and the Agent Team with id 6789 and pkey maps 6666 and 7777.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>Agent Agent Team Member</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <res:NameValuePair>
                     <res:Name>ParentId</res:Name>
                     <res:Value>6789</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>ChildId</res:Name>
                     <res:Value>1234</res:Value>
                  </res:NameValuePair>
```

```
                  <res:NameValuePair>
                     <res:Name>Supervisor</res:Name>
                     <res:Value>false</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>PrimarySupervisor</res:Name>
                     <res:Value>false</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>PhysicalMember</res:Name>
                     <res:Value>true</res:Value>
                  </res:NameValuePair>
               </res:Fields>
               <res:EquipmentMapping>
                  <res:Equipment>
                     <res:Type>Agent Agent Team Pkey</Type>
                     <res:Fields>
                       <res:NameValuePair>
                          <res:Name>ParentId</res:Name>
                          <res:Value>6666</res:Value>
                       </res:NameValuePair>
                       <res:NameValuePair>
                          <res:Name>ChildId</res:Name>
                          <res:Value>1111</res:Value>
                       </res:NameValuePair>
                        <res:NameValuePair>
                           <res:Name>ClusterResourceId</Name>
                           <res:Value>{c5ed5103-55cc-420b-8fa9-9d64c2eaa339}
</Value>
                       </res:NameValuePair>
                     </res:Fields>
                  </res:Equipment>
                  <res:Equipment>
                     <res:Type>Agent Agent Team Pkey</Type>
                     <res:Fields>
                       <res:NameValuePair>
                          <res:Name>ParentId</res:Name>
                          <res:Value>7777</res:Value>
                       </res:NameValuePair>
                       <res:NameValuePair>
                          <res:Name>ChildId</res:Name>
                          <res:Value>2222</res:Value>
                       </res:NameValuePair>
```

```
                        <res:NameValuePair>
                           <res:Name>ClusterResourceId</Name>
                           <res:Value>{5103c5ed-d43e-fa4e-98f0-2eaa33a9d64c}
</Value>
                        </res:NameValuePair>
                     </res:Fields>
                  </res:Equipment>
               </res:EquipmentMapping>
            </res:Resource>
         </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

4.9.43 Create Precision Attribute

4.9.43.1 Soap Example

This example creates a Precision Attribute “Mortgages” in the folder with `FolderId 00000000-0000-0000-0000-000000000010`, where the Attribute is a proficiency attribute (code 4) with a default value of 1.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Create>
         <res:resources>
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>IT_PRECISION_ATTRIBUTE</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00</res:EffectiveTo>
               <res:Status>R</res:Status>
               <res:Fields>
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>00000000-0000-0000-0000-000000000010</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Name</res:Name>
                     <res:Value>Mortgages</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>InternalName</res:Name>
```

```
                  <res:Value>CICM.Mortgages</res:Value>
               </res:NameValuePair>
               <res:NameValuePair>
                  <res:Name>AttributeDataType</res:Name>
                  <res:Value>4</res:Value>
               </res:NameValuePair>
               <res:NameValuePair>
                  <res:Name>DefaultValue</res:Name>
                  <res:Value>1</res:Value>
               </res:NameValuePair>
            </res:Fields>
         </res:Resource>
      </res:resources>
   </res:Create>
</soap:Body>
</soap:Envelope>
```

4.9.44 Create Precision Queue

4.9.44.1 Description

This example creates a Precision Queue, which contains:

- a single Precision Queue Step
- a Precision Queue Step Precision Attribute Member that links the Precision Queue Step to an existing Precision Attribute with id 3333.

The details for the step and the member that links the step and the attribute are specified in the same request as the queue since they must be created at the same time.

When the queue is created, the following items are also created automatically as part of the creation process:

- the specified step
- the specified member that links the step and the attribute
- a Precision Queue Step Precision Queue Member linking the new step with the new queue.

4.9.44.2 Use Case Sequence

Prerequisites:

- The specified Precision Attribute exists

Table 4.180 Create Precision Queue

Action	API Calls and Parameters
Create the Precision Queue and at least one related Prevision Queue Step (which in turn must contain at least one Precision Attribute membership), specifying the details for all three items as in the SOAP example below.	`create PrecisionQueue`
Wait for the asynchronous notification if a subscription has been set up (SOAP only), or begin a poll sequence using search or retrieve.	`search item:<Id> type:<ResourceType>` or `retrieve Id = PrecisionQueueURN resourceType:<ResourceType>`

4.9.44.3 SOAP Example

This SOAP request creates a Precision Queue **Precision Queue 1**, containing one step **Precision Queue 1 Step 1** that links to the Attribute with id **5555**.

The comparison value for this attribute in this step is **7** (field `Value1`), and the comparison to be used is "greater than or equal to" (specified by a value of **6** in field `AttributeRelation`).

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Create>
         <res:resources>
            <!—- Precision Queue -->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>IT_PRECISION_QUEUE</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00</res:EffectiveTo>
               <res:Status>R</res:Status>
                <res:Fields>
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>00000000-0000-0000-0000-000000000010</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Name</res:Name>
                     <res:Value>Precision Queue 1</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>InternalName</res:Name>
```

```
            <res:Value>CICM.PQ1</res:Value>
         </res:NameValuePair>
         <res:NameValuePair>
            <res:Name>AgentOrdering</res:Name>
            <res:Value>1</res:Value>
         </res:NameValuePair>
         <res:NameValuePair>
            <res:Name>ServiceLevelThreshold</res:Name>
            <res:Value>1</res:Value>
         </res:NameValuePair>
         <res:NameValuePair>
            <res:Name>ServiceLevelType</res:Name>
            <res:Value>1</res:Value>
         </res:NameValuePair>
      </res:Fields>
   </res:Resource>
   <!-- First Precision Queue Step -->
   <res:Resource>
       <res:Identity>-1</res:Identity>
       <res:Type>IT_PRECISION_QUEUE_STEP</res:Type>
      <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
      <res:EffectiveTo>2079-06-06T00:00:00</res:EffectiveTo>
      <res:Status>R</res:Status>
       <res:Fields>
         <res:NameValuePair>
            <res:Name>FolderId</res:Name>
            <res:Value>00000000-0000-0000-0000-000000000010</res:Value>
         </res:NameValuePair>
         <res:NameValuePair>
            <res:Name>Name</res:Name>
            <res:Value>Precision Queue 1 Step 1</res:Value>
         </res:NameValuePair>
         <res:NameValuePair>
            <res:Name>StepOrder</res:Name>
            <res:Value>1</res:Value>
         </res:NameValuePair>
         <res:NameValuePair>
            <res:Name>WaitTime</res:Name>
            <res:Value>-1</res:Value>
         </res:NameValuePair>
       <!—- The parent precision queue is being created
              at the same time as this step -->
         <res:NameValuePair>
            <res:Name>PrecisionQueueUrn</res:Name>
            <res:Value>-3</res:Value>
         </res:NameValuePair>
```

```
            </res:Fields>
          </res:Resource>
          <!-- Precision Queue Step Precision Queue Attribute Member,
               Comparison is "Attribute id 5555 >= 7" -->
          <res:Resource>
          <!-- This membership is being created at the same time
                  as the parent step -->
            <res:Identity>-3</res:Identity>
            <res:Type>MT_PRECISION_QUEUE_STEP_PRECISION_ATTRIBUTE_
MEMBER</res:Type>
            <res:Fields>
              <res:NameValuePair>
                 <res:Name>ParentId</res:Name>
                 <res:Value>-3</res:Value>
              </res:NameValuePair>
              <res:NameValuePair>
                 <res:Name>ChildId</res:Name>
                 <res:Value>5555</res:Value>
              </res:NameValuePair>
              <res:NameValuePair>
                 <res:Name>Value1</res:Name>
                 <res:Value>7</res:Value>
              </res:NameValuePair>
              <res:NameValuePair>
                 <res:Name>TermOrder</res:Name>
                 <res:Value>1</res:Value>
              </res:NameValuePair>
              <res:NameValuePair>
                 <res:Name>ParenthesesCount</res:Name>
                 <res:Value>0</res:Value>
              </res:NameValuePair>
              <res:NameValuePair>
                 <res:Name>TermRelation</res:Name>
                 <res:Value>0</res:Value>
              </res:NameValuePair>
              <res:NameValuePair>
                 <res:Name>AttributeRelation</res:Name>
                 <res:Value>6</res:Value>
              </res:NameValuePair>
            </res:Fields>
          </res:Resource>
        </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

4.9.45 Update Precision Queue (Add Precision Queue Step With Two Precision Attributes)

4.9.45.1 Description

This example adds a Precision Queue Step containing two Precision Attribute expressions to the end of an existing Precision Queue.

The details for the members that link the step and the attributes are specified in the same request as the step since they must be created at the same time.

When the step is created, the member linking the new step with the parent queue is also created automatically as part of the creation process.

After the new step is added, the `WaitTime` field in the original step must be modified as that step is no longer the last step in the queue. In this example, a `ConsiderIf` expression is also added to the original step.

This example uses the `Save()` method since the create and update actions must be done in the same SOAP request (the `ConsiderIf` could be updated later, but the `WaitTime` must be updated as the new step is added, because this must be a non-zero value for all steps except the last one).

4.9.45.2 SOAP Example

This SOAP request creates a new Precision Queue Step **Precision Queue Step 2** in the Precision Queue with id **123456**, that links to the Precision Attributes with id **5555** and **6666**.

The comparison value for attribute **5555** in this step is **4** (field `Value1` in the step to attribute membership), and the comparison to be used is "greater than or equal to" (specified by a value of **6** in field `AttributeRelation` in the step to attribute membership).

The comparison value for attribute **6666** in this step is **true** (field `Value1`), and the comparison to be used is "equal to" (specified by a value of **1** in field `AttributeRelation`). The `TermRelation` field has a value of **2**, indicating that this attribute term is to be compared with the previous one using an "OR" operation.

This step is the new last step in the queue, so the existing step, with id **6789** is updated to have a wait time of 20 seconds.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Save>
         <res:resourcesToAdd>
```

```
            <!—Add new Precision Queue Step, second step in Precision Queue -->
            <res:Resource>
                <res:Identity>-1</res:Identity>
                <res:Type>IT_PRECISION_QUEUE_STEP</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T00:00:00</res:EffectiveTo>
               <res:Status>R</res:Status>
                <res:Fields>
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>00000000-0000-0000-0000-000000000010</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Name</res:Name>
                     <res:Value>Precision Queue 1 Step 2</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>StepOrder</res:Name>
                     <res:Value>2</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>WaitTime</res:Name>
                     <res:Value>-1</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>PrecisionQueueUrn</res:Name>
                     <res:Value>123456</res:Value>
                  </res:NameValuePair>
               </res:Fields>
             </res:Resource>
            <!—- Add Precision Queue Step Precision Queue Attribute Member 1
                 Comparison is “Attribute id 5555 >= 4” -->
             <res:Resource>
               <!—- This membership will be created at the same time
                    as the parent step -->
                <res:Identity>-3</res:Identity>
                <res:Type>MT_PRECISION_QUEUE_STEP_PRECISION_ATTRIBUTE_
MEMBER</res:Type>
                <res:Fields>
                  <res:NameValuePair>
                     <res:Name>ParentId</res:Name>
                     <res:Value>-3</res:Value>
                  </res:NameValuePair>
```

```
                  <res:NameValuePair>
                     <res:Name>ChildIdId</res:Name>
                     <res:Value>5555</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Value1</res:Name>
                     <res:Value>4/res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>TermOrder</res:Name>
                     <res:Value>1</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>ParenthesesCount</res:Name>
                     <res:Value>0/res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>TermRelation</res:Name>
                     <res:Value>0</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>AttributeRelation</res:Name>
                     <res:Value>6</res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
           <!—- Add Precision Queue Step Precision Queue Attribute Member 2,
                Comparison is “OR Attribute id 6666 = true” -->
            <res:Resource>
              <!—- This membership will be created at the same time
                   as the parent step -->
               <res:Identity>-3</res:Identity>
               <res:Type>MT_PRECISION_QUEUE_STEP_PRECISION_ATTRIBUTE_
MEMBER</res:Type>
               <res:Fields>
                 <res:NameValuePair>
                    <res:Name>ParentId</res:Name>
                    <res:Value>-3</res:Value>
                 </res:NameValuePair>
                 <res:NameValuePair>
                    <res:Name>ChildIdId</res:Name>
                    <res:Value>6666</res:Value>
                 </res:NameValuePair>
```

```
                  <res:NameValuePair>
                     <res:Name>Value1</res:Name>
                     <res:Value>true/res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>TermOrder</res:Name>
                     <res:Value>2</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>ParenthesesCount</res:Name>
                     <res:Value>0/res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>TermRelation</res:Name>
                     <res:Value>2</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>AttributeRelation</res:Name>
                     <res:Value>1</res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
           <!—Update existing Precision Queue Step with new WaitTime -->
           <res:Resource>
               <res:Identity>6789</res:Identity>
               <res:Type>IT_PRECISION_QUEUE_STEP</res:Type>
               <res:Fields>
                 <res:NameValuePair>
                    <res:Name>WaitTime</res:Name>
                    <res:Value>20</res:Value>
                 </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resourcesToAdd>
      </res:Save>
   </soap:Body>
</soap:Envelope>
```

4.9.46 Update Precision Queue (Delete Precision Queue Step)

4.9.46.1 Description

This example deletes the first Precision Queue Step from an existing Precision Queue that contained two Precision Queue Steps.

When the step is deleted, the following items are also deleted automatically:

- the member linking the deleted step with the related attribute
- the member linking the deleted step with the parent queue.

In this example, the `StepOrder` field of the remaining step needs to be modified as it is now the first step in the queue.

This example uses the `Save()` method since the delete and update actions must be done in the same SOAP request (the `StepOrder` for the remaining step must be updated as the other step is deleted).

4.9.46.2 SOAP Example

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Save>
        <res:resourcesToDelete>
            <!—- Delete reqired Precision Queue Step -->
          <res:Resource>
              <res:Identity>6789</res:Identity>
              <res:Type>IT_PRECISION_QUEUE_STEP</res:Type>
           </res:Resource>
         </res:resourcesToDelete>
         <res: resourcesToAdd>
            <!-- Update StepOrder for existing Precision Queue Step -->
            <res:Resource>
              <res:Identity>6790</res:Identity>
              <res:Type>IT_PRECISION_QUEUE_STEP</res:Type>
              <res:Fields>
                <res:NameValuePair>
                   <res:Name>StepOrder</res:Name>
                   <res:Value>1</res:Value>
                </res:NameValuePair>
              </res:Fields>
           </res:Resource>
          </res:resourcesToAdd>
      </res:Update>
   </soap:Body>
</soap:Envelope>
```

5 Subscriptions Web Service

5.1 About the Subscriptions Web Service

The Subscriptions Web Service allows the client application to subscribe to receive a notification when a specified item changes state in Cisco Unified CCE.

When a specified change occurs, the client receives a notification about the change. The notification includes the identity, name and status of the item.

5.2 Subscriptions Web Service Specification

5.2.1 SOAP

The WSDL file for the Subscriptions Web Service is located on the Web/Application server here:

https://<server>:8083/ResourceManagement?wsdl

where **<server>** is the name of the Web/Application server.

The Subscriptions Web Service implements the WS-Eventing interface which is specified at **http://www.w3.org/Submission/WS-Eventing/**.

5.2.2 REST

The Subscriptions Web Service does not support the REST protocol.

5.3 Securing Notifications with SSL

5.3.1 About Securing the Subscriptions Web Service

This section describes how to use SSL to secure and authenticate the communications from the Unified CCDM web server (the server generating the notifications) to the subscriber server (the server running the subscriber client application)

This procedure is optional. It is only necessary if you plan to use the Unified CCDM Subscriptions Web Service, and want to secure and authenticate the communications between the Unified CCDM server and the subscriber server. If you are not using the Unified CCDM Subscriptions Web Service, or if you do not

need to secure and authenticate communications from the Unified CCDM server to the subscriber server (for example, if all your servers are co-located in a secure area) then this procedure is not required.

Note
In order to use the Unified CCDM Web Services, you must first secure the Web Services themselves as described in the *Installation and Configuration Guide for Cisco Unified Contact Center Domain Manager*, section *Securing Unified CCDM Web Service APIs with SSL*. This procedure is not optional and you will not be able to use the Unified CCDM Web Services unless you have done this.

When Unified CCDM is first installed, communications between the Unified CCDM server and the subscriber server are secured with a self-signed certificate called *localhost*. This certificate is suitable for a single server in a laboratory environment, but cannot be used to secure a multi-server installation in a production environment.

To secure and authenticate the communications between the Unified CCDM server and the subscriber server:

- obtain and install a suitable digital certificate on the Unified CCDM server (see section 5.3.2 "Obtaining and Installing a Digital Certificate for the Unified CCDM Server")
- obtain and install a suitable digital certificate on the subscriber server (see section 5.3.3 "Obtaining and Installing a Digital Certificate for the Subscriber Server")
- configure the Unified CCDM server endpoint behavior to use the two certificates for communication between the servers (see section 5.3.4 "Configure the Unified CCDM Server Endpoint Behavior")
- install the public key of the subscriber server on the Unified CCDM server (see section 5.3.5 "Installing the Public Key of the Subscriber Server Certificate onto the Unified CCDM Server")
- ensure the certificate authority root certificate for the Unified CCDM server certificate is available on the subscriber server (see section 5.3.6 "Installing the Root CA Certificate on the Subscriber server")
- restart the services on both servers.

5.3.2 Obtaining and Installing a Digital Certificate for the Unified CCDM Server

This step installs the digital certificate that secures the Unified CCDM side of the communications between the Unified CCDM App/Web server and the subscriber server.

Note
When Unified CCDM was installed, the Unified CCDM Web Services should have been secured with SSL (see *Installation and Configuration Guide for Cisco Unified Contact Center Domain Manager*, section *Securing Unified CCDM Web Service APIs with SSL*). If this has not been done, complete that procedure before continuing with the instructions here.

As part of the process of securing the Unified CCDM Web Services with SSL, a digital certificate was obtained and installed on the Unified CCDM App/Web server. We recommend that you use the same digital certificate here. If you choose to use the same certificate, then it has already been installed and this step is complete.

If you want to use a different certificate, follow the instructions in *Installation and Configuration Guide for Cisco Unified Contact Center Domain Manager* to obtain and install another certificate on the Unified CCDM App/Web server and to grant access to the **NETWORK SERVICE** user.

5.3.3 Obtaining and Installing a Digital Certificate for the Subscriber Server

This step installs the digital certificate that secures the subscriber server side of the communications between the Unified CCDM App/Web server and the subscriber server.

You may already have a suitable certificate that you can use to secure the subscriber server side of the communications between the Unified CCDM App/Web server and the subscriber server. If you do not already have a suitable certificate, consult your platform documentation to find out how to request or generate an external certificate (suitable for public use) or an internal certificate (for secure use within the issuing organization).

When you have obtained the certificate, install it on the subscriber server as described in your platform documentation.

Ensure that the account that will be used to run the subscriber application has full access rights for the certificate. Again, consult your platform documentation if necessary.

5.3.4 Configure the Unified CCDM Server Endpoint Behavior

This step configures the endpoint behavior of the Unified CCDM App/Web server to use the server certificates to secure communications between the two servers.

To do this:

1. Identify the *subject distinguished name* of the certificate you have installed on the Unified CCDM App/Web server. This is a text string, containing several name-value pairs, that identifies the certificate, for example:

   `"CN=gold, OU=dev, O=exony, L=newbury, S=berkshire, C=GB"`

   If you do not know the subject distinguished name, you can find it as follows:

   a. On the Unified CCDM server, go to the **Start** menu, and type **mmc** in the command box to open Microsoft Management Console (MMC).

   b. Click **File > Add/Remove Snap-in**, click **Certificates**, then **Add.**

   c. In the Certificates Snap-in dialog box, select **Computer Account** and click **Next**.

   d. In the Select Computer dialog box, select **Local Computer** and click **Finish** to add the Certificates snap-in to MMC. Click **OK**.

   e. In MMC, expand the Certificates node and the Personal node, then click **Certificates** to see the available certificates.

   f. Double-click on the certificate and in the **Details** tab, select **Subject**, to see the name-value pairs that make up the subject distinguished name for the certificate. Note that your certificate may not have exactly the same name-value pairs as the example above.

2. Identify the subject distinguished name of the certificate you have installed on the subscriber server. If the subscriber server is a Windows platform you can follow the instructions above. Otherwise, consult your platform documentation to find out how to do this.

3. On the Unified CCDM App/Web server, go to **C:\Program Files\Domain Manager\Application Server**, locate the file **Exony.Reporting.Application.Server.exe.config** and open it in a text editor.

Tip

You may want to make a safe copy of this file before you change it.

4. Locate the section that begins

   `<behavior name="ClientCertificateBehavior">`

   and within that section, the section that begins

   `<clientCredentials>`

5. In the `<clientCredentials>` section, locate the `<clientCertificate>` tag and change the `findValue` attribute to the subject distinguished name of the

Unified CCDM server certificate. Leave the other attributes as they are. Using the example above, the contents of the <clientCertificate> tag may become:

```
<clientCertificate findValue="CN=gold, OU=dev,O=exony, L=newbury,
S=berkshire, C=GB" storeLocation="LocalMachine"
storeName="My"
x509FindType="FindBySubjectDistinguishedName"
/>
```

6. Also in the <clientCredentials> section, locate the <serviceCertificate> tag, and below that, the <defaultCertificate> tag. Set the findValue attribute of the <defaultCertificate>tag to the subject distinguished name of the subscriber server certificate. Leave the other attributes as they are. For example, if the subject distinguished name of the certificate was

   ```
   "CN=silver, OU=dev, O=exony, L=newbury, S=berkshire, C=GB"
   ```

   the contents of the <defaultCertificate> tag may become:

   ```
   <defaultCertificate findValue="CN=silver, OU=dev, O=exony,
   L=newbury, S=berkshire, C=GB"
   storeLocation="LocalMachine" storeName="My"
   x509FindType="FindBySubjectDistinguishedName" />
   ```

7. Save the changes to this file.

5.3.5 Installing the Public Key of the Subscriber Server Certificate onto the Unified CCDM Server

This step ensures that the Unified CCDM App/Web server can encrypt the communications in the way required by the subscriber server.

To install the public key of the subscriber server certificate onto the Unified CCDM App/Web server:

1. On the subscriber server, obtain the public key of the subscriber server certificate and save it to a file. Consult your platform documentation to find out how to do this.
2. Copy the file containing the public key to the Unified CCDM App/Web server.
3. On the Unified CCDM App/Web server, start MMC, expand the Certificates node and the Personal node, then right-click Certificates. Select **All Tasks > Import**. Follow the steps in the Certificate Import Wizard to import the file containing the subscriber server public key into the Personal certificate store of the local machine.

5.3.6 Installing the Root CA Certificate on the Subscriber server

This step confirms the validity of the Unified CCDM web services certificate to the subscriber server. This is done by obtaining the root certificate of the certificate authority (CA) that issued the Unified CCDM web services certificate and installing it on the subscriber server.

Note

This step is only necessary if the provider of the Unified CCDM server certificate is not already trusted by the subscriber server. If the certificate was purchased from one of the major certificate providers, then the root certificate of the CA will almost certainly be pre-loaded on the subscriber server. In this case, the CA is already trusted.

To install the CA root certificate for the Unified CCDM server certificate:

1. Obtain the CA root certificate for the Unified CCDM server certificate. If you did not get this when you purchased the certificate, you can obtain it from the certificate authority. Copy the file containing the CA root certificate to the subscriber server.
2. On the subscriber server, import the CA root certificate into the Trusted Root certificate area. Consult your platform documentation to find out how to do this if necessary.

5.3.7 Restarting the Services

1. Save all configuration files and other changes.
2. On the Unified CCDM App/Web server, stop and restart the Unified CCDM Web Service.
3. On the subscriber server, stop and restart the application service.

5.4 Subscriptions Web Service APIs

5.4.1 Subscribe

5.4.1.1 Description

The `Subscribe` web service API creates a subscription for item or member changes.

5.4.1.2 Example

Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <wse:Subscribe xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
        <wse:Delivery>
          <wse:NotifyTo>
            <a:Address> https://localhost:33333/Sink </a:Address>
          </wse:NotifyTo>
        </wse:Delivery>
        <wse:Expires>PT15M</wse:Expires>
        <wse:Filter xmlns:x="http://www.exony.com/portal">//x:Id = 1234 and
//x:Type = 'Agent'</wse:Filter>
      </wse:Subscribe>
  </s:Body></s:Envelope>
```

In this example, a subscription has been created for **Agent** items with an ID of **1234**.

- The `Filter` element can contain any valid XPath query which is applied to the incoming status message (described in more detail later).
- The `Expires` element defines how long the subscription is held by the server. In this example, **T15M** indicates that the subscription is held for 15 minutes.
- The `Delivery` element contains the `NotifyTo` node which holds an `Address`. This is the SOAP endpoint that implements the `IStatus` web service interface.

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
...</s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <wse:SubscribeResponse
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
         <wse:SubscriptionManager>
            <a:Address>https://localhost:8083/SubscriptionManager</a:Address>
            <a:ReferenceParameters>
               <wse:Identifier>uuid:ecf6f297-6fce-4924-a973-
aeb533378ef6</wse:Identifier>
            </a:ReferenceParameters>
         </wse:SubscriptionManager>
         <wse:Expires>2009-12-24T10:33:22.9328168Z</wse:Expires>
      </wse:SubscribeResponse>
   </s:Body>
</s:Envelope>
```

The example response shows the returned **Identifier**. This can be used with the other Web Service methods to identify this subscription.

The **Expires** element contains the absolute expiry date time for this subscription.

5.4.2 Get Status

5.4.2.1 Description

The `GetStatus` web service API allows the client to check the status of a subscription.

5.4.2.2 Example

Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <h:Identifier xmlns:h="http://schemas.xmlsoap.org/ws/2004/08/eventing"
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">uuid:67f4901f-e183-4ea2-
83cb-24f07e82d3e9</h:Identifier>
  </s:Header>
```

```
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <wse:GetStatus
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing"></wse:GetStatus>
  </s:Body>
</s:Envelope>
```

Response

```
s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://schemas.xmlsoap.org/ws/2004/08/eventing/GetStatusRespon
se</a:Action>
      <a:RelatesTo>uuid:1128b40a-f9a7-46f0-8ca5-67aa6d410607</a:RelatesTo>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2009-12-24T10:31:12.187Z</u:Created>
            <u:Expires>2009-12-24T10:36:12.187Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <wse:GetStatusResponse
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
         <wse:Expires>2009-12-24T10:40:10.3298445Z</wse:Expires>
      </wse:GetStatusResponse>
   </s:Body>
</s:Envelope>
```

5.4.3 Renew

5.4.3.1 Description

The `Renew` web service API allows the client to extend the lifetime of a subscription.

5.4.3.2 Example

Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <h:Identifier xmlns:h="http://schemas.xmlsoap.org/ws/2004/08/eventing"
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">uuid:67f4901f-e183-4ea2-
83cb-24f07e82d3e9</h:Identifier>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <wse:Renew xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
      <wse:Expires>PT20M</wse:Expires>
    </wse:Renew>
  </s:Body>
</s:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <wse:RenewResponse
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
         <wse:Expires>2009-12-24T10:51:35.8457528Z</wse:Expires>
      </wse:RenewResponse>
   </s:Body>
</s:Envelope>
```

5.4.4 Unsubscribe

5.4.4.1 Description

The `Unscribe` web service API allows the client to cancel an existing subscription.

5.4.4.2 Example

Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <h:Identifier xmlns:h="http://schemas.xmlsoap.org/ws/2004/08/eventing"
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">uuid:67f4901f-e183-4ea2-
83cb-24f07e82d3e9</h:Identifier>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <wse:Unsubscribe
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing"></wse:Unsubscribe>
  </s:Body>
</s:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"/>
</s:Envelope>
```

5.5 More Information about Subscriptions and Notifications

5.5.1 Subscribing for Notifications

When the client subscribes for notifications, the client specifies an XPath query as the filter to choose the event messages to receive. The XPath query is applied to this message.

The filter supports the use of the following XPath components:

- `//x:Id`
  The identity of the item to subscribe to.
- `//x:Type`
  The item type of items to subscribe to (for example, Agent, Agent Team).
- `//x:Status`
  The status type that the subscription is concerned with. For example a client may be interested in receiving a notification when an agent becomes ready or is deleted.
- `//x:ContainerId`
  The identity of the folder in which items must belong. This may be used to monitor multiple items for a given folder, for example, to receive a notification when any agent within a particular folder is updated.

The client can define a subscription query by combining one or more of these XPath components. For example:

- `//x:Id = 1234` and `//x:Type = 'Agent'` will cause a notification to be received by the client when the agent with identity 1234 changes state.
- `//x:Type = 'Agent'` and `//x:Status = 'R'` and `//x:ContainerId = ‘1BAE1951-A9FF-4F17-AC46-FA7605C26569'` will cause a notification to be sent when any agent in the folder with the identity 1BAE1951-A9FF-4F17-AC46-FA7605C26569 changes status to R.

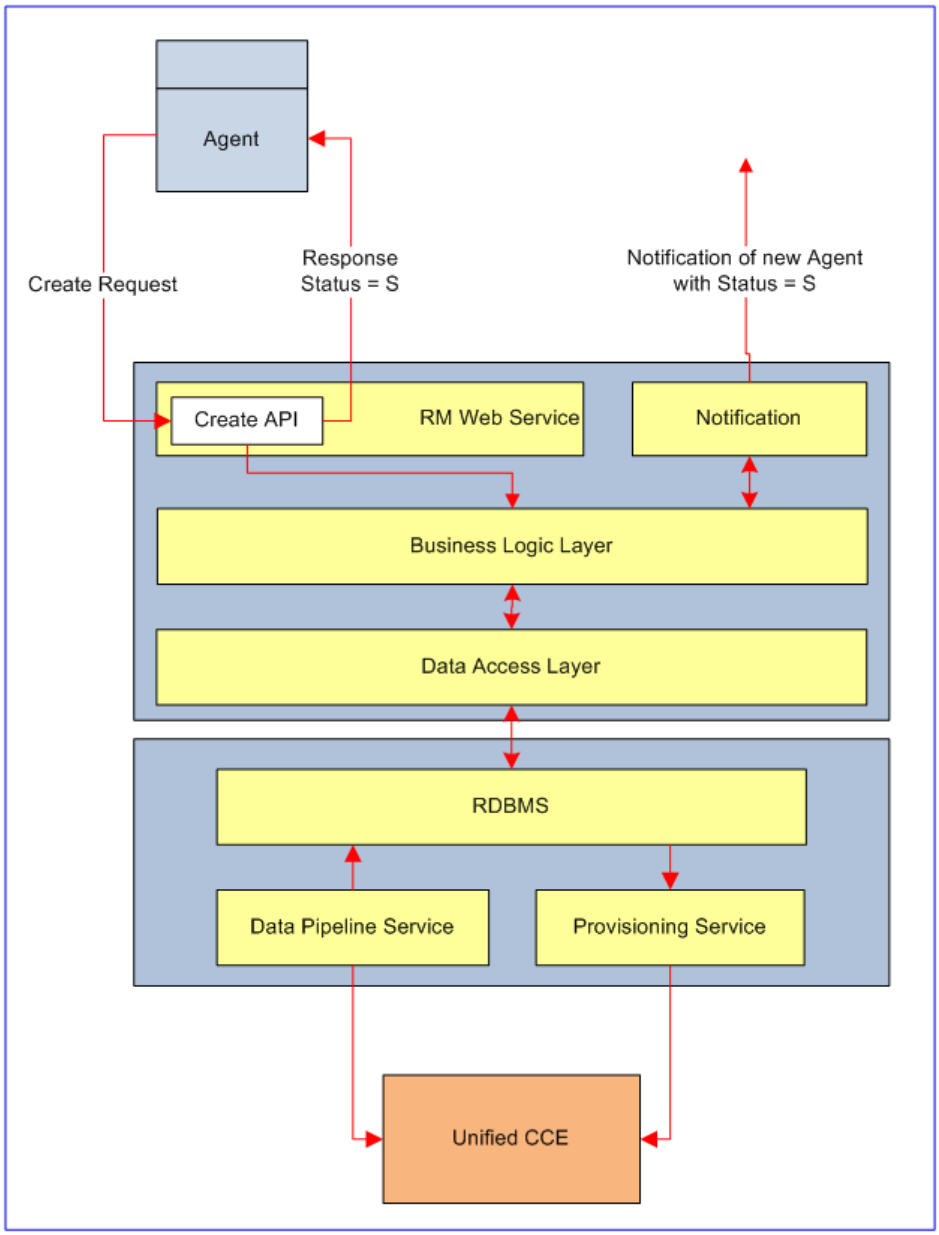
Notifications are only received for items that support status changes. If an API request (for example `Create()`) returns a status of Ready then the status of that item is not updated and no notification will be received.

5.5.2 Notification Reporting

Certain resource types require Unified CCDM to remotely provision them. These resources will return a Pending or Synchronize state when fields and memberships are added, updated or deleted. The client can subscribe to these items to allow status changes to be obtained as the item moves through the provisioning lifecycle.

An example work flow for the agent creation process is as follows:

1. Agent create request is performed by the client, business logic is applied to the item and it is written to the Unified CCDM database waiting to be provisioned onto the remote equipment (in this case Unified CCE). Notification is sent out to all subscribers to say that a new agent has been added and its status is synchronizing.
2. The Unified CCDM Provisioning Server picks up the new change and attempt to make the change on Unified CCE.
3. The Unified CCE successfully creates the new agent, the provisioning server updates the item status in the RDBMS and all subscribers receive a notification that the agent has progressed to Ready status (that is, has been created on Unified CCE).



Not all resource types use the provisioning life-cycle. Some resources, for example, User, Group etc are not provisioned on any remote equipment. They are added to Unified CCDM immediately. Therefore the notification behavior of these types will be subtly different than those that utilize the provisioning life-cycle detailed above.

The following types of notification are supported by Unified CCDM:

- Support Provisioning Life-cycle – has the behavior described above, these are types that rely on Unified CCDM to provision them on remote equipment. These types will progress through the provisioning life-cycle and report their progress at each stage using the notifications to report their current state.
- Returns Status on W/S Call - returns an accurate status of the resource where no provisioning life-cycle exists. This method is used for items that are not required to be provisioned on remote equipment for example users/ groups etc. The status is reported at the time of the operation. An error status indicates that the operation was not successful and the resource will remain in its existing state.
- Notification on Create - when a new resource is created a notification request is received containing the new item identity and the new item status.
- Notification on Update - when a resource is updated then a notification request is received containing the updated item's identity and status.
- Notification on Delete - when a resource is deleted then a notification request is received containing the identity of the item and its new status.

The following table shows the supported types and the level of notification support that they adopt.

Type	Support Provisioning Life-cycle	Returns Status on W/S Call	Notification on Create	Notification on Update	Notification on Delete
Agent	X		X	X	X
Agent Team	X		X	X	X
Call Type	X		X	X	X
Folder		X	X		X
Group		X	X		X
Person	X		X	X	X
Skill Group	X		X	X	X
User		X	X		X
Agent Agent Desktop Member	X		X		X
Agent Agent Team Member	X		X	X	X

Type	Support Provisioning Life-cycle	Returns Status on W/S Call	Notification on Create	Notification on Update	Notification on Delete
Agent Skill Group Member	X		X		X
User Group Member		X	X		
Group Group Member		X	X		

5.5.3 Handling Notification Failures

Subscription and notification information is not persisted in the event of a failure. Notifications are sent periodically when a status change is identified for any subscribed resource, for example, when AgentA moves from status R to status S. Given the distributed nature of the platform it is possible that an item may progress through more than one state before the notification change is collected.

The following subscription method should therefore be applied to protect client applications when notifications are not received (for example, in the event of a failure).

1. Subscribe to resource x (currently status = R)
2. Perform change (for example, Call Update API, status = S)
3. Await ready notification
4. If notification is not received in 30 seconds call Search for the item identity of resource x
5. If status = R then progress, if status = S then go back to step 3

This technique provides a fail-safe tracking mechanism for resources progressing through the provisioning life-cycle.

5.5.4 API Notification Timeout

5.5.4.1 Description

It is possible that notifications could be lost (for example, if there are network problems) so if it is critical for a response to be received, then we suggest that the caller implements a timer to retrieve the result.

5.5.4.2 Sequence Diagram

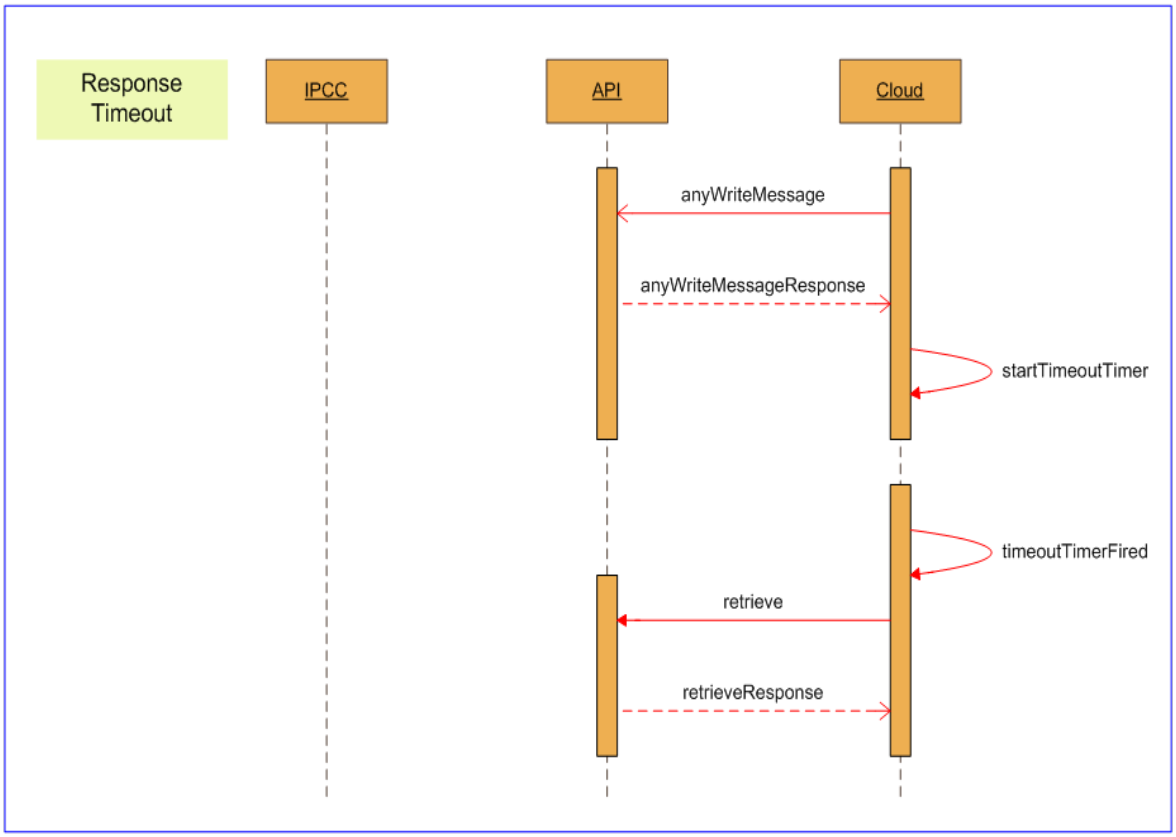


Figure 5.1 API Notification Timeout Sequence Diagram

5.5.5 Closed Loop Poll

5.5.5.1 Description

Moves, adds and changes may be made in the underlying remote equipment. Unified CCDM periodically reads the configuration data from the remote equipment and updates the Unified CCDM data model accordingly. Updating the data model can have two results:

- Resource notifications will be sent to any clients who have subscribed to them.

- Additional Unified CCDM provisioning workflows may be triggered. For example, the addition of a CUCM Line with an IPCC category set in the description field will trigger the addition of a Device Target / Labels per Routing Client on the relevant Unified CCE. Provisioning workflows may trigger additional status changes.

5.5.5.2 Sequence Diagram

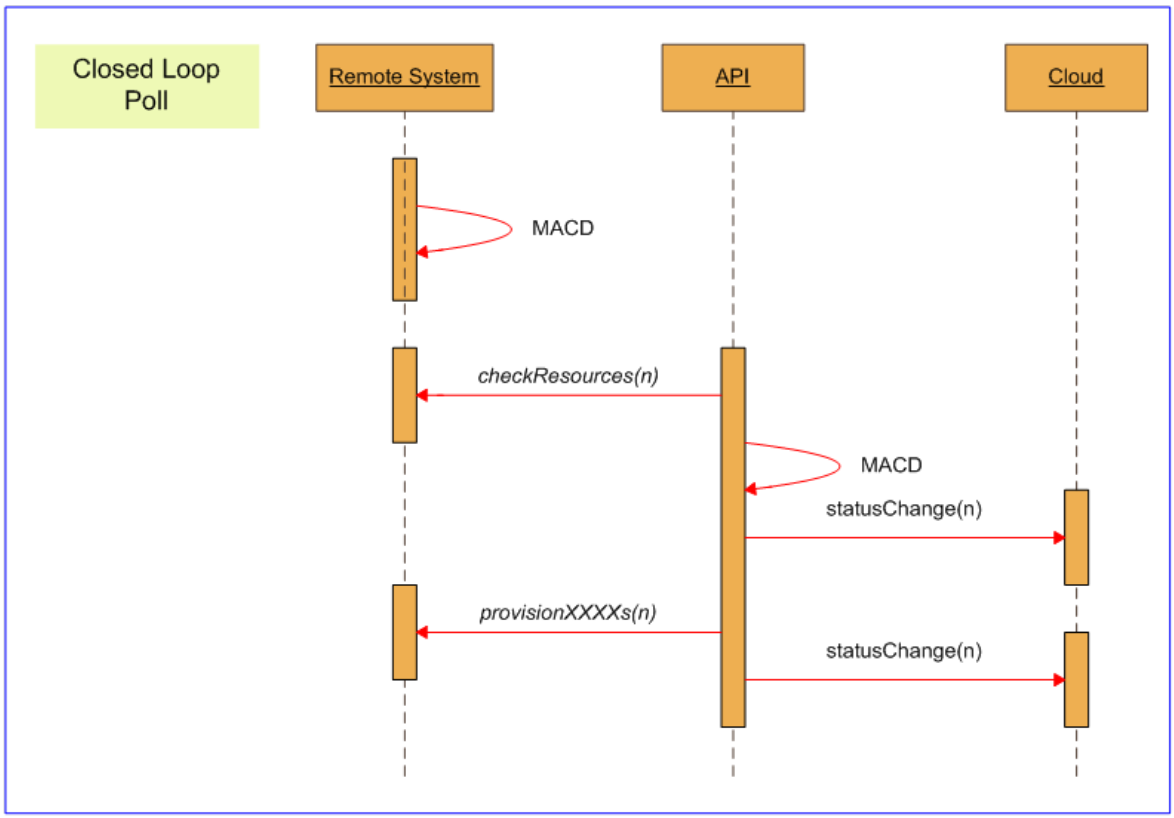


Figure 5.2 Closed Loop Poll Sequence Diagram

5.5.6 Interaction with Resource Management Web Services

See section 5.6 "Subscriptions Web Service Examples" for some detailed examples showing how the Subscriptions and Resource Management Web Services interact to complete some common tasks.

5.6 Subscriptions Web Service Examples

5.6.1 Agent Re-Skilling

A typical agent re-skilling activity may be performed using the following requests and response sequence.

5.6.1.1 Step 1

Load the agent that is to be re-skilled using the Resource Management `Retrieve()` web service API.

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org//www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Retrieve>
         <!--Optional:-->
         <res:resourceKeys>
            <!--Zero or more repetitions:-->
            <res:ResourceKey>
               <res:Identity>17585</res:Identity>
               <res:ResourceType>Agent</res:ResourceType>
            </res:ResourceKey>
         </res:resourceKeys>
      </res:Retrieve>
   </soap:Body>
</soap:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
 <s:Header>
      <a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/RetrieveResponse</a:Action>
      <a:RelatesTo>uuid:701c0aa3-12d1-4728-b199-aae95d64514a</a:RelatesTo>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2010-01-19T13:57:54.911Z</u:Created>
            <u:Expires>2010-01-19T14:02:54.911Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
      <RetrieveResponse
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
         <RetrieveResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <Resource>
               <Identity>17585</Identity>
               <Type>Agent</Type>
               <EffectiveFrom>2010-01-18T08:58:36</EffectiveFrom>
               <EffectiveTo>2079-06-06T00:00:00</EffectiveTo>
               <Status>R</Status>
               <Changestamp>0</Changestamp>
               <Fields>
                  <NameValuePair>
                     <Name>FolderId</Name>
                     <Value>1bae1951-a9ff-4f17-ac46-fa7605c26569</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>Name</Name>
                     <Value>V723.PG1_1.USER67.AG</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>InternalName</Name>
                     <Value>V723.PG1_1.USER67.AG</Value>
                  </NameValuePair>
```

```
<NameValuePair>
   <Name>Description</Name>
   <Value>User67</Value>
</NameValuePair>
<NameValuePair>
   <Name>PeripheralUrn</Name>
   <Value>3529</Value>
</NameValuePair>
<NameValuePair>
   <Name>PersonUrn</Name>
   <Value>9409</Value>
</NameValuePair>
<NameValuePair>
   <Name>FirstName</Name>
   <Value>User67</Value>
</NameValuePair>
<NameValuePair>
   <Name>LastName</Name>
   <Value>User67</Value>
</NameValuePair>
<NameValuePair>
   <Name>PeripheralNumber</Name>
   <Value>50067</Value>
</NameValuePair>
<NameValuePair>
   <Name>PeripheralName</Name>
   <Value/>
</NameValuePair>
<NameValuePair>
   <Name>Supervisor</Name>
   <Value>False</Value>
</NameValuePair>
<NameValuePair>
   <Name>AgentStateTrace</Name>
   <Value>False</Value>
</NameValuePair>
<NameValuePair>
   <Name>DomainName</Name>
   <Value/>
</NameValuePair>
<NameValuePair>
   <Name>DomainLoginName</Name>
   <Value/>
</NameValuePair>
```

```
                <NameValuePair>
                   <Name>DomainUserName</Name>
                   <Value/>
                </NameValuePair>
                <NameValuePair>
                   <Name>DomainUserGuid</Name>
                   <Value/>
                </NameValuePair>
                <NameValuePair>
                   <Name>DomainPassPhrase</Name>
                   <Value/>
                </NameValuePair>
                <NameValuePair>
                   <Name>UserDeletable</Name>
                   <Value>True</Value>
                </NameValuePair>
                <NameValuePair>
                   <Name>UserGroupId</Name>
                   <Value/>
                </NameValuePair>
             </Fields>
          </Resource>
       </RetrieveResult>
    </RetrieveResponse>
  </s:Body>
</s:Envelope>
```

5.6.1.2 Step 2

Subscribe to receive a notification when the status of the agent that is to be re-skilled changes.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<wse:Subscribe xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
<wse:Delivery>
<wse:NotifyTo>
<a:Address> https://localhost:33333/Sink </a:Address>
</wse:NotifyTo>
</wse:Delivery>
<wse:Expires>PT15M</wse:Expires>
<wse:Filter xmlns:x="http://www.exony.com/portal">//x:Id = 17585 and //x:Type =
'Agent'</wse:Filter>
```

```
</wse:Subscribe>
</s:Body>
</s:Envelope>
```

5.6.1.3 Step 3

Load the list of Skill Groups in which the Agent already exists so that they can be displayed for removal and may be excluded from the list of available Skill Groups. The list of available Skill Groups can be retrieved using the Resource Management `Search()` API.

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Search>
         <res:queryString>type:SkillGroup parentof:Agent,17585</res:queryString>
         <res:excludeFilter></res:excludeFilter>
      </res:Search>
   </soap:Body>
</soap:Envelope>
```

Notes

Replace the selected identity in this request with the identity of the agent retrieved in Step 1.

Response

The response contains a collection of Skill Group resources for the Skill Groups that are mapped to that agent.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/SearchResponse</a:Action>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2010-01-19T14:24:22.545Z</u:Created>
            <u:Expires>2010-01-19T14:29:22.545Z</u:Expires>
         </u:Timestamp>
      </o:Security>
```

```
   </s:Header>
   <s:Body>
      <SearchResponse
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
         <SearchResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <Resource>
               <Identity>17321</Identity>
               <Type>SkillGroup</Type>
               <EffectiveFrom>2010-01-18T09:04:51</EffectiveFrom>
               <EffectiveTo>2079-06-06T00:00:00</EffectiveTo>
               <Status>R</Status>
               <Changestamp>0</Changestamp>
               <Fields>
                  <NameValuePair>
                     <Name>MediaRoutingDomainUrn</Name>
                     <Value>8377</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>PeripheralUrn</Name>
                     <Value>3529</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>FolderId</Name>
                     <Value>1bae1951-a9ff-4f17-ac46-fa7605c26569</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>Name</Name>
                     <Value/>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>InternalName</Name>
                     <Value>AvayaPG_1.Cisco_Voice.Avaya2.pri</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>Description</Name>
                     <Value/>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>PeripheralNumber</Name>
                     <Value>2</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>PeripheralName</Name>
                     <Value>AvayaSkill2.pri</Value>
                  </NameValuePair>
                  <NameValuePair>
```

```
            <Name>AvailableHoldoffDelay</Name>
            <Value>-1</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>Priority</Name>
            <Value>1</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>Extension</Name>
            <Value/>
         </NameValuePair>
         <NameValuePair>
            <Name>IPTA</Name>
            <Value>N</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>ServiceLevelThreshold</Name>
            <Value>0</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>ServiceLevelType</Name>
            <Value>0</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>DefaultEntry</Name>
            <Value>0</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>SubSkillGroupMask</Name>
            <Value/>
         </NameValuePair>
         <NameValuePair>
            <Name>UserDeletable</Name>
            <Value>True</Value>
         </NameValuePair>
      </Fields>
   </Resource>
   <Resource>
      <Identity>17329</Identity>
      <Type>SkillGroup</Type>
      <EffectiveFrom>2010-01-18T09:04:51</EffectiveFrom>
      <EffectiveTo>2079-06-06T00:00:00</EffectiveTo>
      <Status>R</Status>
      <Changestamp>0</Changestamp>
      <Fields>
         <NameValuePair>
```

```
      <Name>MediaRoutingDomainUrn</Name>
      <Value>8377</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>PeripheralUrn</Name>
      <Value>3529</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>FolderId</Name>
      <Value>1bae1951-a9ff-4f17-ac46-fa7605c26569</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>Name</Name>
      <Value/>
   </NameValuePair>
   <NameValuePair>
      <Name>InternalName</Name>
      <Value>AvayaPG_1.Cisco_Voice.Avaya3.pri</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>Description</Name>
      <Value/>
   </NameValuePair>
   <NameValuePair>
      <Name>PeripheralNumber</Name>
      <Value>3</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>PeripheralName</Name>
      <Value>AvayaSkill3.pri</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>AvailableHoldoffDelay</Name>
      <Value>-1</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>Priority</Name>
      <Value>1</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>Extension</Name>
      <Value/>
   </NameValuePair>
   <NameValuePair>
      <Name>IPTA</Name>
      <Value>N</Value>
```

```
                    </NameValuePair>
                    <NameValuePair>
                       <Name>ServiceLevelThreshold</Name>
                       <Value>0</Value>
                    </NameValuePair>
                    <NameValuePair>
                       <Name>ServiceLevelType</Name>
                       <Value>0</Value>
                    </NameValuePair>
                    <NameValuePair>
                       <Name>DefaultEntry</Name>
                       <Value>0</Value>
                    </NameValuePair>
                    <NameValuePair>
                       <Name>SubSkillGroupMask</Name>
                       <Value/>
                    </NameValuePair>
                    <NameValuePair>
                       <Name>UserDeletable</Name>
                       <Value>True</Value>
                    </NameValuePair>
                 </Fields>
              </Resource>
           </SearchResult>
        </SearchResponse>
     </s:Body>
</s:Envelope>
```

Note

This agent is assigned to 2 skill groups. The highlighted skill group identities are used in the next step to exclude them from the list of available selections.

5.6.1.4 Step 4

Load the list of Skill Groups that the agent can be added to using the Resource Management `Search()` API. The Skill Groups that have already been mapped are excluded from the results so that they are not available for selection.

The Resource Management `Search()` API is used to retrieve all Skill Groups on a Peripheral. This uses the `PeripheralUrn` from the agent retrieved in step 1 (agents may only be skilled to skill groups on the same peripheral).

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Search>
         <res:queryString>type:SkillGroup childof:Peripheral,3529</res:queryString>
         <res:excludeFilter>
            <res:ResourceKey>
               <res:Identity>17321</res:Identity>
               <res:ResourceType>SkillGroup</res:ResourceType>
            </res:ResourceKey>
            <res:ResourceKey>
               <res:Identity>17329</res:Identity>
               <res:ResourceType>SkillGroup</res:ResourceType>
            </res:ResourceKey>
         </res:excludeFilter>
      </res:Search>
   </soap:Body>
</soap:Envelope>
```

Response

The response from this request returns all the skill groups on the peripheral, except the skill groups with identities 17321 and 17329.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/SearchResponse</a:Action>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2010-01-19T14:40:09.175Z</u:Created>
            <u:Expires>2010-01-19T14:45:09.175Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
      <SearchResponse
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
         <SearchResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <Resource>
```

```
<Identity>17241</Identity>
<Type>SkillGroup</Type>
<EffectiveFrom>2010-01-18T09:04:49</EffectiveFrom>
<EffectiveTo>2079-06-06T00:00:00</EffectiveTo>
<Status>R</Status>
<Changestamp>0</Changestamp>
<Fields>
   <NameValuePair>
      <Name>MediaRoutingDomainUrn</Name>
      <Value>8377</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>PeripheralUrn</Name>
      <Value>3529</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>FolderId</Name>
      <Value>1bae1951-a9ff-4f17-ac46-fa7605c26569</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>Name</Name>
      <Value/>
   </NameValuePair>
   <NameValuePair>
      <Name>InternalName</Name>
      <Value>PG1_1.Cisco_Voice.default.72576</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>Description</Name>
      <Value/>
   </NameValuePair>
   <NameValuePair>
      <Name>PeripheralNumber</Name>
      <Value>19882</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>PeripheralName</Name>
      <Value>000474608774</Value>
   </NameValuePair>
   <NameValuePair>
      <Name>AvailableHoldoffDelay</Name>
      <Value>-1</Value>
   </NameValuePair>
   <NameValuePair>
```

```
            <Name>Priority</Name>
            <Value>0</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>Extension</Name>
            <Value/>
         </NameValuePair>
         <NameValuePair>
            <Name>IPTA</Name>
            <Value>Y</Value>
        </NameValuePair>
         <NameValuePair>
            <Name>ServiceLevelThreshold</Name>
            <Value>0</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>ServiceLevelType</Name>
            <Value>0</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>DefaultEntry</Name>
            <Value>1</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>SubSkillGroupMask</Name>
            <Value/>
         </NameValuePair>
         <NameValuePair>
           <Name>UserDeletable</Name>
            <Value>True</Value>
         </NameValuePair>
      </Fields>
   </Resource>
   <Resource>
      <Identity>17249</Identity>
      <Type>SkillGroup</Type>
      <EffectiveFrom>2010-01-18T09:04:49</EffectiveFrom>
      <EffectiveTo>2079-06-06T00:00:00</EffectiveTo>
      <Status>R</Status>
      <Changestamp>0</Changestamp>
      <Fields>
         <NameValuePair>
            <Name>MediaRoutingDomainUrn</Name>
            <Value>8377</Value>
         </NameValuePair>
```

```
<NameValuePair>
   <Name>PeripheralUrn</Name>
   <Value>3529</Value>
</NameValuePair>
<NameValuePair>
   <Name>FolderId</Name>
   <Value>1bae1951-a9ff-4f17-ac46-fa7605c26569</Value>
</NameValuePair>
<NameValuePair>
   <Name>Name</Name>
   <Value/>
</NameValuePair>
<NameValuePair>
   <Name>InternalName</Name>
   <Value>Skill2</Value>
</NameValuePair>
<NameValuePair>
   <Name>Description</Name>
   <Value/>
</NameValuePair>
<NameValuePair>
   <Name>PeripheralNumber</Name>
   <Value>1</Value>
</NameValuePair>
<NameValuePair>
   <Name>PeripheralName</Name>
   <Value>Skill2</Value>
</NameValuePair>
<NameValuePair>
   <Name>AvailableHoldoffDelay</Name>
   <Value>-1</Value>
</NameValuePair>
<NameValuePair>
   <Name>Priority</Name>
   <Value>0</Value>
</NameValuePair>
<NameValuePair>
   <Name>Extension</Name>
   <Value/>
</NameValuePair>
<NameValuePair>
   <Name>IPTA</Name>
   <Value>Y</Value>
</NameValuePair>
<NameValuePair>
```

```
                  <Name>ServiceLevelThreshold</Name>
                  <Value>0</Value>
               </NameValuePair>
               <NameValuePair>
                  <Name>ServiceLevelType</Name>
                  <Value>0</Value>
               </NameValuePair>
               <NameValuePair>
                  <Name>DefaultEntry</Name>
                  <Value>0</Value>
               </NameValuePair>
               <NameValuePair>
                  <Name>SubSkillGroupMask</Name>
                  <Value/>
               </NameValuePair>
               <NameValuePair>
                  <Name>UserDeletable</Name>
                  <Value>True</Value>
               </NameValuePair>
            </Fields>
         </Resource>
      </SearchResult>
   </SearchResponse>
</s:Body>
</s:Envelope>
```

5.6.1.5 Step 5

Add the Agent to a Skill Group. The Resource Management `Create()` API is called passing the identity of the Agent and Skill Group to create a new Agent Skill Group Member.

Request

Here is the agent skill group member creation request. `ParentId` describes the identity of the skill group to which the agent is being associated. `ChildId` describes the identity of the agent that is being added to the skill group.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Create>
         <res:resources>
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>Agent SkillGroup Member</res:Type>
```

```
            <res:EffectiveFrom>1900-01-01T00:00:00</res:EffectiveFrom>
            <res:EffectiveTo>2079-06-06T23:59:59.9999999</res:EffectiveTo>
            <res:Status>S</res:Status>
            <res:Changestamp>0</res:Changestamp>
            <res:Fields>
               <res:NameValuePair>
                  <res:Name>ParentId</res:Name>
                  <res:Value>17249</res:Value>
               </res:NameValuePair>
               <res:NameValuePair>
                  <res:Name>ChildId</res:Name>
                  <res:Value>17585</res:Value>
               </res:NameValuePair>
            </res:Fields>
         </res:Resource>
      </res:resources>
   </res:Create>
</soap:Body>
</soap:Envelope>
```

Response

The identity in the response is the identity of the Agent Skill Group Member object that is being removed.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/CreateResponse</a:Action>
      <a:RelatesTo>uuid:e1b93eaf-2ba8-4a68-8a66-8aaded6524a8</a:RelatesTo>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2010-01-19T15:09:18.902Z</u:Created>
            <u:Expires>2010-01-19T15:14:18.902Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
      <CreateResponse
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
         <CreateResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <RequestResult>
```

```
            <Identity>14757</Identity>
            <Status>S</Status>
            <Errors i:nil="true"/>
          </RequestResult>
        </CreateResult>
      </CreateResponse>
   </s:Body>
</s:Envelope>
```

5.6.1.6 Step 6

Notification received that the Agent has returned to ready (R) status.

Notification of an agent change is retrieved first when the re-skilling change is committed informing that the status has changed to Synchronizing (S) and then again once the change has been committed and the item has progressed to Ready (R) status.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
<s:Header></s:Header>
<s:Body>
 <OnStatusChange xmlns="http://www.exony.com/schemas/2009/10/status">
<status xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<Id>17585</Id>
<Status>R</Status>
<Type>Agent</Type>
<ContainerId>1bae1951-a9ff-4f17-ac46-fa7605c26569</ContainerId>
</status>
</OnStatusChange>
 </s:Body>
</s:Envelope>
```

5.6.1.7 Step 7

Delete the Agent from a Skill Group

Once the agent returns to Ready (R) status the Resource Management `Delete()` API is called passing the identity of the Agent Skill Group Member that is to be removed.

To delete and Agent from a Skill Group we need to obtain the identity of the Agent Skill Group Member object that maps the two items together.

This can be achieved using the following search query in the Resource Management Search API to retrieve all of the Agent Skill Group Member objects for skill groups mapped to this agent:

```
memberbychild:Agent,17585,"Agent Skill Group Member"
```

The identity of the Agent Skill Group Member that is to be removed should then be used in the Resource Management `Delete` web service API. The request for this API will look like this:

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Delete>
         <res:resourceKeys>
            <res:ResourceKey>
              <res:Identity>14757</res:Identity>
               <res:ResourceType>Agent SkillGroup Member</res:ResourceType>
           </res:ResourceKey>
         </res:resourceKeys>
      </res:Delete>
   </soap:Body>
</soap:Envelope>
```

This will produce a response similar to this:

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/DeleteResponse</a:Action>
      <a:RelatesTo>uuid:c4a43490-1bd8-4856-8bbf-8516f56a1904</a:RelatesTo>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2010-01-19T15:32:58.487Z</u:Created>
            <u:Expires>2010-01-19T15:37:58.487Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
      <DeleteResponse
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
         <DeleteResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <RequestResult>
               <Identity>14757</Identity>
```

```
            <Status>P</Status>
            <Errors i:nil="true"/>
          </RequestResult>
        </DeleteResult>
      </DeleteResponse>
   </s:Body>
</s:Envelope>
```

5.6.1.8 Step 8

Notification received that the Agent has returned to ready (R) status.

Notification of an agent change is retrieved first when the re-skilling change is committed informing that the status has changed to Synchronizing (S) and then again once the change has been committed and the item has progressed to Ready (R) status.

Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
<s:Header></s:Header>
<s:Body>
 <OnStatusChange xmlns="http://www.exony.com/schemas/2009/10/status">
<status xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <Id>17585</Id>
    <Status>R</Status>
    <Type>Agent</Type>
                          <ContainerId>1bae1951-a9ff-4f17-ac46-
fa7605c26569</ContainerId>
</status>
</OnStatusChange>
 </s:Body>
</s:Envelope>
```

5.6.2 Agent Creation

The following steps must be performed for a typical agent creation activity:

A typical agent creation activity may be performed using the following requests and response sequence.

5.6.2.1 Step 1

Create a Person.

Create a person record using the Resource Management `Create()` API. The XML request for this will look similar to this.

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
     <res:Create>
         <!--Optional:-->
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Identity>-1</res:Identity>
               <res:Type>Person</res:Type>
               <res:EffectiveFrom>0001-01-01T00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06T23:59:59.9999999</res:EffectiveTo>
               <res:Status>S</res:Status>
               <res:Changestamp>0</res:Changestamp>
               <res:Fields>
                  <!--Zero or more repetitions:-->
                  <res:NameValuePair>
                     <res:Name>FolderId</res:Name>
                     <res:Value>221C6722-B830-4848-9521-35B2DD8757D7</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Name</res:Name>
                     <res:Value>JohnSmith</res:Value>
                  </res:NameValuePair
                 <res:NameValuePair>
                     <res:Name>FirstName</res:Name>
                     <res:Value>John</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>LastName</res:Name>
                     <res:Value>Smith</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>Description</res:Name>
                     <res:Value>A description</res:Value>
                  </res:NameValuePair>
                  <res:NameValuePair>
                     <res:Name>LoginName</res:Name>
                     <res:Value>john.smith</res:Value>
                  </res:NameValuePair>
```

```
                  <res:NameValuePair>
                     <res:Name>PassPhrase</res:Name>
                     <res:Value>Pa55word</res:Value>
                  </res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

This request will return the identity of the new person record which we will need later on. In this example the identity of the person returned is 21365. This is returned from the web service in the following XML structure:

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<s:Header>
<a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/CreateResponse</a:Action>
<o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
<u:Timestamp u:Id="_0">
<u:Created>2010-01-19T11:29:22.429Z</u:Created>
<u:Expires>2010-01-19T11:34:22.429Z</u:Expires>

</u:Timestamp>
</o:Security>
</s:Header>
<s:Body>
<CreateResponse xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
<CreateResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><RequestResult>
<Identity>21365</Identity>
<Status>S</Status>
<Errors i:nil="true"/>
</RequestResult>
</CreateResult>
</CreateResponse>
</s:Body>
</s:Envelope>
```

5.6.2.2 Step 2

Subscribe to the Person.

After creation the returned identity should immediately be subscribed to so that when the status is updated to ready, the associated agent may be created. The subscription request must be made to the Subscriptions `Subscribe()` method using an XML request as follows:

Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<wse:Subscribe xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
<wse:Delivery>
<wse:NotifyTo>
<a:Address> https://localhost:33333/Sink </a:Address>
</wse:NotifyTo>
</wse:Delivery>
<wse:Expires>PT15M</wse:Expires>
<wse:Filter xmlns:x="http://www.exony.com/portal">//x:Id = 21365 and //x:Type =
'Person'</wse:Filter>
</wse:Subscribe>
</s:Body>
</s:Envelope>
```

The Address element must be updated to an endpoint that implements the `IStatus` web service interface and the filter must include the person identity and type from the above request. The expiry of this notification is set to 15 minutes.

> **Note**
> It is possible that the item may have changed status between the create and subscription request. Therefore it is recommended that the Resource Management Search API is called after the subscription is setup to ensure that the status has not already changed from synchronizing (S).

5.6.2.3 Step 3

Receive the Person subscription notification.

Within the 15 minute subscription period a notification will be sent to the subscriber when the associated item has changed. This notification will be an XML request in the following format:

Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
<s:Header></s:Header>
<s:Body>
<OnStatusChange xmlns="http://www.exony.com/schemas/2009/10/status">
<status xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<Id>21365</Id>
<Status>R</Status>
<Type>Person</Type>
<ContainerId>221C6722-B830-4848-9521-35B2DD8757D7</ContainerId>
</status>
</OnStatusChange>
</s:Body>
</s:Envelope>
```

5.6.2.4 Step 4

Create the Agent.

When the notification has been received and the created person has moved into the ready status (R), the agent creation request can be sent to the Resource Management `Create()` API.

The following XML request creates the agent associating the newly created person:

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Create>
         <res:resources>
            <!--Zero or more repetitions:-->
            <res:Resource>
               <res:Type>Agent</res:Type>
               <res:EffectiveFrom>2009-01-01 00:00:00</res:EffectiveFrom>
               <res:EffectiveTo>2079-06-06 00:00:00</res:EffectiveTo>
               <res:Status>S</res:Status>
               <res:Fields>
<res:NameValuePair>
<res:Name>FolderId</res:Name>
<res:Value>9F44B644-7C24-40CD-9DB9-6D1175DD7FE3</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>PeripheralUrn</res:Name>
```

```
<res:Value>-1</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>PersonUrn</res:Name>
<res:Value>21365</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>Name</res:Name>
<res:Value>Agent1</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>InternalName</res:Name>
<res:Value>CICM.Agent1</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>Description</res:Name>
<res:Value>Description of Agent1</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>FirstName</res:Name>
<res:Value>Jim</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>LastName</res:Name>
<res:Value>Smith</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>PeripheralNumber</res:Name>
<res:Value>132456</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>PeripheralName</res:Name>
<res:Value>Agent1</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>Supervisor</res:Name>
<res:Value>True</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>DomainName</res:Name>
<res:Value>DOMAIN</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>DomainLoginName</res:Name>
<res:Value>DOMAIN\Agent1</res:Value>
</res:NameValuePair>
<res:NameValuePair>
```

```
      <res:Name>DomainUserName</res:Name>
<res:Value>Agent1</res:Value>
</res:NameValuePair>
<res:NameValuePair>
      <res:Name>DomainUserGuid</res:Name>
<res:Value>66965475b1d6a448aafc8fb6deac2b2f</res:Value>
</res:NameValuePair>
               </res:Fields>
            </res:Resource>
         </res:resources>
      </res:Create>
   </soap:Body>
</soap:Envelope>
```

The response for the above request will return the identity of the new agent.

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<s:Header>
<a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/CreateResponse</a:Action>
<o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
<u:Timestamp u:Id="_0">
<u:Created>2010-01-19T11:29:22.429Z</u:Created>
<u:Expires>2010-01-19T11:34:22.429Z</u:Expires>
</u:Timestamp>
</o:Security>
</s:Header>
<s:Body>
<CreateResponse xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
<CreateResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><RequestResult>
<Identity>21369</Identity>
<Status>S</Status>
<Errors i:nil="true"/>
</RequestResult>
</CreateResult>
</CreateResponse>
</s:Body>
</s:Envelope>
```

5.6.2.5 Step 5

Subscribe to the Agent using the new identity.

Subscribe to the agent so that a notification is sent when the agent is updated. This will allow the end user to be informed once the change has been made to the remote system. The following XML request should be sent to the Subscriptions `Subscribe()` API:

Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<wse:Subscribe xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
<wse:Delivery>
<wse:NotifyTo>
<a:Address> https://localhost:33333/Sink </a:Address>
</wse:NotifyTo>
</wse:Delivery>
<wse:Expires>PT15M</wse:Expires>
<wse:Filter xmlns:x="http://www.exony.com/portal">//x:Id = 21369 and //x:Type =
'Agent'</wse:Filter>
</wse:Subscribe>
</s:Body>
</s:Envelope
```

5.6.2.6 Step 6

Receive the Agent Subscription Notification.

Within the 15 minute subscription period a notification will be sent to the subscriber when the associated item has changed. This notification will be an XML request in the following format:

Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
<s:Header></s:Header>
<s:Body>
<OnStatusChange xmlns="http://www.exony.com/schemas/2009/10/status">
<status xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<Id>21369</Id>
<Status>R</Status>
<Type>Agent</Type>
<ContainerId>9F44B644-7C24-40CD-9DB9-6D1175DD7FE3</ContainerId>
</status>
</OnStatusChange>
</s:Body>
</s:Envelope>
```

Once this notification is received then the agent has been successfully created and may be skilled or added to an agent team.

Note

Whilst these examples demonstrate a solution for a single item, it may be considered more efficient to setup and maintain a recurring subscription to a given type. For example, if the Subscribe API is called specifying just a type and not an identity then notifications will be sent for all changes to all items of that type.

5.6.3 Dialed Number From Unallocated folder

Resources that are not allocated to a tenant or shared across multiple tenants are held in the equipment specific sub folder under the **/Unallocated** folder. A typical example of such a resource is the Dialed Number resource type.

To using the Resource Management Web Services to obtain a list of unallocated Dialed Numbers:

5.6.3.1 Step 1

Use the Resource Management `Search()` API to obtain the identity of the folder containing unallocated Dialed Numbers for this Unified CCE:

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
<soap:Header/>
   <soap:Body>
      <res:Search>
         <res:queryString>type:Folder folder:"00000002-0000-0000-0000-000000000005"
name:ProvisioningCICM</res:queryString>
      </res:Search>
   </soap:Body>
</soap:Envelope>
```

Note
The identity of the unallocated folder used in the above search query is fixed for all Unified CCDM installations.

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/SearchResponse</a:Action>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2010-03-12T16:47:35.520Z</u:Created>
            <u:Expires>2010-03-12T16:52:35.520Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
      <SearchResponse
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
         <SearchResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <Resource>
               <Identity>dc68cb53-4654-49aa-a947-24592fd1bc39</Identity>
               <Type>Folder</Type>
               <EffectiveFrom>0001-01-01T00:00:00</EffectiveFrom>
               <EffectiveTo>2079-06-06T23:59:59.9999999</EffectiveTo>
               <Status>R</Status>
               <Changestamp>0</Changestamp>
```

```
               <Fields>
                  <NameValuePair>
                     <Name>FolderId</Name>
                     <Value>00000002-0000-0000-0000-000000000005</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>Name</Name>
                     <Value>ProvisioningCICM</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>Description</Name>
                     <Value>Unallocated resources for equipment [ProvisioningCICM]
</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>Hidden</Name>
                     <Value>False</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>Path</Name>
                     <Value>/Unallocated/ProvisioningCICM</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>PolicyId</Name>
                     <Value>00000002-0000-0000-0000-000000000004</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>FolderType</Name>
                     <Value>F</Value>
                  </NameValuePair>
               </Fields>
            </Resource>
         </SearchResult>
      </SearchResponse>
   </s:Body>
</s:Envelope>
```

Note

Record the identity of the returned resource as this is used in the next query to retrieve the list of unallocated Dialed Numbers.

5.6.3.2 Step 2

Call the Resource Management `Search()` API using the retrieved Folder ID to request the list of unallocated Dialed numbers for this Unified CCE.

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:res="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <res:Search>
         <res:queryString>type:"Dialed Number" folder:"dc68cb53-4654-49aa-a947-
24592fd1bc39" latest:1 status:R max:10</res:queryString>
      </res:Search>
   </soap:Body>
</soap:Envelope>
```

The list of Dialed Numbers contained within the unallocated folder is returned to the client:

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/SearchResponse</a:Action>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2010-03-12T16:59:02.937Z</u:Created>
            <u:Expires>2010-03-12T17:04:02.937Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
      <SearchResponse
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
         <SearchResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <Resource>
               <Identity>31081</Identity>
               <Type>Dialed Number</Type>
               <EffectiveFrom>2010-03-12T16:32:46.857</EffectiveFrom>
               <EffectiveTo>2079-06-06T00:00:00</EffectiveTo>
               <Status/>
```

```
      <Changestamp>0</Changestamp>
      <Fields>
         <NameValuePair>
            <Name>FolderId</Name>
            <Value>dc68cb53-4654-49aa-a947-24592fd1bc39</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>Name</Name>
            <Value/>
         </NameValuePair>
         <NameValuePair>
            <Name>InternalName</Name>
            <Value>V723.JONO.DN</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>Description</Name>
            <Value/>
         </NameValuePair>
         <NameValuePair>
            <Name>Digits</Name>
            <Value>465456456</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>PermitApplicationRouting</Name>
            <Value>False</Value>
         </NameValuePair>
      </Fields>
   </Resource>
   <Resource>
      <Identity>31085</Identity>
      <Type>Dialed Number</Type>
      <EffectiveFrom>2010-03-12T16:32:46.96</EffectiveFrom>
      <EffectiveTo>2079-06-06T00:00:00</EffectiveTo>
      <Status/>
      <Changestamp>0</Changestamp>
      <Fields>
         <NameValuePair>
            <Name>FolderId</Name>
            <Value>dc68cb53-4654-49aa-a947-24592fd1bc39</Value>
         </NameValuePair>
         <NameValuePair>
            <Name>Name</Name>
            <Value/>
         </NameValuePair>
         <NameValuePair>
            <Name>InternalName</Name>
            <Value>V723.FG.DN</Value>
```

```
      </NameValuePair>
      <NameValuePair>
         <Name>Description</Name>
         <Value/>
      </NameValuePair>
      <NameValuePair>
         <Name>Digits</Name>
         <Value>456</Value>
      </NameValuePair>
      <NameValuePair>
         <Name>PermitApplicationRouting</Name>
         <Value>False</Value>
      </NameValuePair>
   </Fields>
</Resource>
<Resource>
   <Identity>31089</Identity>
   <Type>Dialed Number</Type>
   <EffectiveFrom>2010-03-12T16:32:46.963</EffectiveFrom>
   <EffectiveTo>2079-06-06T00:00:00</EffectiveTo>
   <Status/>
   <Changestamp>0</Changestamp>
   <Fields>
      <NameValuePair>
         <Name>FolderId</Name>
         <Value>dc68cb53-4654-49aa-a947-24592fd1bc39</Value>
      </NameValuePair>
      <NameValuePair>
         <Name>Name</Name>
         <Value/>
      </NameValuePair>
      <NameValuePair>
         <Name>InternalName</Name>
         <Value>V723.DN45.DN</Value>
      </NameValuePair>
      <NameValuePair>
         <Name>Description</Name>
         <Value/>
      </NameValuePair>
      <NameValuePair>
         <Name>Digits</Name>
         <Value>436436745</Value>
      </NameValuePair>
      <NameValuePair>
         <Name>PermitApplicationRouting</Name>
         <Value>False</Value>
      </NameValuePair>
```

```
            </Fields>
         </Resource>
      </SearchResult>
   </SearchResponse>
</s:Body>
</s:Envelope>
```

Note

When returning large sets of resources, you can use the `Max` and `Offset` keywords with the `Search()` API to return pages of resources.

5.6.3.3 Step 3

Once the list is obtained then unallocated Dialed Numbers should be moved to the associated tenant folder when they are to be allocated. This is achieved using the Resource Management `Update()` API, passing in the details of the item to be moved and the associated folder ids as follows:

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
   <soap:Header/>
   <soap:Body>
      <Update>
        <resources>
            <Resource>
               <Identity>31089</Identity>
               <Type>Dialed Number</Type>
               <EffectiveFrom>2010-03-11T12:19:37</EffectiveFrom>
               <EffectiveTo>2011-03-12T16:32:45.96</EffectiveTo>
               <Status>R</Status>
               <Changestamp>0</Changestamp>
               <Fields>
                  <NameValuePair>
                     <Name>FolderId</Name>
                     <Value>DC68CB53-4654-49AA-A947-24592FD1BC39</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>NewFolderId</Name>
                     <Value>240253C4-3F30-48C5-A3E6-65B3D94B100E</Value>
                  </NameValuePair>
                  <NameValuePair>
                     <Name>Name</Name>
                     <Value>V723.DN45.DN</Value>
                  </NameValuePair>
                  <NameValuePair>
```

```
                  <Name>InternalName</Name>
                  <Value>V723.DN45.DN</Value>
               </NameValuePair>
               <NameValuePair>
                  <Name>Description</Name>
                  <Value/>
               </NameValuePair>
               <NameValuePair>
                  <Name>RoutingClientUrn</Name>
                  <Value>22049</Value>
               </NameValuePair>
               <NameValuePair>
                  <Name>MediaRoutingDomainUrn</Name>
                  <Value>10789</Value>
               </NameValuePair>
               <NameValuePair>
                  <Name>Digits</Name>
                  <Value>436436745</Value>
               </NameValuePair>
               <NameValuePair>
                  <Name>PermitApplicationRouting</Name>
                  <Value>False</Value>
               </NameValuePair>
            </Fields>
         </Resource>
      </resources>
   </Update>
</soap:Body>
</soap:Envelope>
```

This moves the dialed number from the folder `DC68CB53-4654-49AA-A947-24592FD1BC39` to the folder `240253C4-3F30-48C5-A3E6-65B3D94B100E`.

When moving items between tenant folders or to or from the unallocated folder, Unified CCDM may delete and recreate the item to enforce security on any associated historical data. This means that the identity for the item will change, and so should be updated in the remote system for future web service calls. The response to the `Update()` request contains the identity of all updated items and allows remote systems storing these keys to be updated.

An example response for the above request where a delete and recreate has been performed is as follows:

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
```

```
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://www.exony.com/schemas/2009/10/resourcemanagement/IResou
rceManagement/UpdateResponse</a:Action>
      <a:RelatesTo>uuid:11c09613-483c-4f1c-9072-b18a7ecfda1b</a:RelatesTo>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2010-03-16T12:31:33.991Z</u:Created>
            <u:Expires>2010-03-16T12:36:33.991Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
      <UpdateResponse
xmlns="http://www.exony.com/schemas/2009/10/resourcemanagement">
         <UpdateResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <RequestResult>
               <Identity>31093</Identity>
               <Status>R</Status>
               <Errors i:nil="true"/>
            </RequestResult>
         </UpdateResult>
      </UpdateResponse>
   </s:Body>
</s:Envelope>
```

Note the new identity and use it for all future web service calls relating to this resource.

Appendix A Unified CCDM Technical Overview

A.1 Unified CCDM Management Overview

Unified CCDM marries contact center platforms to business requirements through a multilayered, partitioned framework. At the user interface level it is designed to empower the numerous user types with the appropriate control and reporting surface. Its core framework provides security, hierarchical partitioning and dual side resilience. The lower levels of the framework provide interfacing and workflow to gather data from the call center components and through workflow provision the components in a very flexible, scripted manner.

Control/Provisioning

- Creation of interaction handling strategies.
- Uploading of IVR speech files.
- Routine maintenance of application.
- Exception handling.

Management Intelligence

- Consolidated cross-platform enterprise reporting.
- Actionable Intelligence.
- Business Relevant.

Business Integration

- Telephony to business system integration.

Framework

- Sophisticated security and partitioning model.
- Core database repository based on RDBMS.
- Modular connector framework for heterogeneous equipment connectivity.
- Thin client user interface.
- Multi-site resilience.
- Integrated Workflow to orchestrate asynchronous operations such as equipment off-line or human interaction requirements.

A.2 Web Service API Architecture

The Unified CCDM Web Services API implementation exposes a number of web services allowing remote provisioning requests to be serviced for third party applications.

These third party applications are typically B2B applications rather than composite client/server interactions.

This first of these services is the Resource Management service, which exposes a web service layer on top of the existing platforms provisioning abstraction layer. Remote requests are serviced via the web service, building low level objects from passed in type-safe parameters. Provisioning changes are then executing using the existing platform provisioning mechanisms committing change to the Cisco Unified CCE environment using their exposed software interfaces. For this service Unified CCDM supports two web service protocols, REST and SOAP for maximum interoperability.

The next of these services is the Subscription Management service, which exposes another web service layer to allow third party applications the ability to subscribe for events when a given resource is changed. This subscription based event service can be used by third party applications to monitor resource items progress through the provisioning life cycle. The REST protocol cannot be used with this web service.

A.3 Web Service API Design Concepts

The Unified CCDM Web Services are hosted within the Unified CCDM application server and do not use the Web Server components at all. Accordingly, they are stateless with a mid-tier cache and distributed for maximum scaling through load balancers. Concurrency checks and two-phase atomic commits (2PC) are implemented in the underlying transactional RDBMS layer for any provisioning write request.

Clients must be aware that each consecutive request can be load balanced to a different server and that the usual race conditions will be present as data makes its way through the systems. For example; a Create call will return an identifier from a Side A server but a subsequent Search/Retrieve on Side B may not see that identifier for up to 10 seconds as the mid-tier caches are updated. Note: this is not a problem in practice since the client already has the new identifier and it can therefore be used in a subsequent provisioning request since write requests are always re-loaded from the RDBMS irrespective of the cache state.

A.4 Performance Tips

Like all secure web service stacks, the excellent interoperability offered by this technology comes at a high performance cost in terms of throughput and latency. There are many reasons for this, including HTTP bloat and network latency .

Some common observations and tips when using the Unified CCDM Web Services:

- Web service latencies are measured in the hundreds to thousands of milliseconds which is significantly slower than the binary streaming operations between App Servers to Databases found in traditional client server systems. Hence, make each call count by requesting or doing as much as possible in each call
- HTTP/S is a best-effort delivery service. This means any request could simply be dropped. Hence, the clients of the Unified CCDM Web Services have to handle this and retry.
- Take the frequency of the messaging into account. Remedy: For provisioning, the majority of data interactions are read-many/write-seldom making them suitable for caching techniques in the higher levels; ideally caching at the edge nearest the users. In the context of many users making many similar small calls then this can be simply implemented as taking the results out of a higher level cache such as memcache.
- High volume aggregation services which involve many elements of state based data queries can be particularly difficult to scale. Hence, replicate key data items into a high level persistent model (the Master Data Source) and use regular re-synchronization, ideally retrieving data during off-hours in large, course-grained transactions.
- Repeated client calls to access server state can choke a network and degrade the server performance. So cache data on the client whenever possible to avoid requests to the server.
- The SOAP protocol is more resource-intensive than the REST equivalent. SOAP has a high intrinsic overhead because it has to extract the SOAP envelope, parse the contained XML information and handle the typing information required in every SOAP message. The XML data cannot be optimized very much, and the binary representation of an object may be expanded by around 500% when encoded in XML.

Appendix B SOAP Messages

B.1 SOAP Request Format

Requests sent to the Unified CCDM Web Services are in the standard SOAP format. The standard SOAP envelope XML format is shown below.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Header/>
   <soap:Body>
       <!-- Message contents here -->
   </soap:Body>
</soap:Envelope>
```

B.2 SOAP Response Format

Responses from the Unified CCDM Web Services are in the standard SOAP format. The standard SOAP envelope XML format is show below.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing"
        xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action s:mustUnderstand="1">?</a:Action>
      <a:RelatesTo>?</a:RelatesTo>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>?</u:Created>
            <u:Expires>?</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <!-- Message contents here -->
   </s:Body>
</s:Envelope>
```

Appendix C Valid Time Zones

The following time zones are valid to be specified in the `TimeZone` field of the User Resource Item (see section 4.4.3 "User Resource Item").

Afghanistan Standard Time
Alaskan Standard Time
Arab Standard Time
Arabian Standard Time
Arabic Standard Time
Argentina Standard Time
Atlantic Standard Time
AUS Central Standard Time
AUS Eastern Standard Time
Azerbaijan Standard Time
Azores Standard Time
Canada Central Standard Time
Cape Verde Standard Time
Caucasus Standard Time
Cen. Australia Standard Time
Central America Standard Time
Central Asia Standard Time
Central Brazilian Standard Time
Central Europe Standard Time
Central European Standard Time
Central Pacific Standard Time
Central Standard Time
Central Standard Time (Mexico)
China Standard Time
Dateline Standard Time
E. Africa Standard Time
E. Australia Standard Time
E. Europe Standard Time
E. South America Standard Time
Eastern Standard Time
Egypt Standard Time
Ekaterinburg Standard Time
Fiji Standard Time
FLE Standard Time
Georgian Standard Time
GMT Standard Time
Greenland Standard Time
Greenwich Standard Time
GTB Standard Time

Hawaiian Standard Time
India Standard Time
Iran Standard Time
Jerusalem Standard Time
Jordan Standard Time
Korea Standard Time
Malay Peninsula Standard Time
Mid-Atlantic Standard Time
Middle East Standard Time
Montevideo Standard Time
Morocco Standard Time
Mountain Standard Time
Mountain Standard Time (Mexico)
Myanmar Standard Time
N. Central Asia Standard Time
Namibia Standard Time
Nepal Standard Time
New Zealand Standard Time
Newfoundland Standard Time
North Asia East Standard Time
North Asia Standard Time
Pacific SA Standard Time
Pacific Standard Time
Pacific Standard Time (Mexico)
Pakistan Standard Time
Romance Standard Time
Russian Standard Time
SA Eastern Standard Time
SA Pacific Standard Time
SA Western Standard Time
Samoa Standard Time
SE Asia Standard Time
South Africa Standard Time
Sri Lanka Standard Time
Taipei Standard Time
Tasmania Standard Time
Tokyo Standard Time
Tonga Standard Time
US Eastern Standard Time
US Mountain Standard Time
Venezuela Standard Time
Vladivostok Standard Time
W. Australia Standard Time
W. Central Africa Standard Time

W. Europe Standard Time
West Asia Standard Time
West Pacific Standard Time
Yakutsk Standard Time

Appendix D Example Notification Endpoint WSDL

This Appendix contains some ex ample WSDL for a subscriber client that expects to receive SOAP requests containing notifications about state changes that the client has previously subscribed to receive.

This is the WSDL implemented by an Unified CCDM test subscriber.

```
 <?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions name="SinkStatus"
targetNamespace="http://www.exony.com/schemas/2009/10/status"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:tns="http://www.exony.com/schemas/2009/10/status"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
<wsp:Policy wsu:Id="TrustedWebClientBinding_IStatus_policy">
- <wsp:ExactlyOne>
- <wsp:All>
- <sp:TransportBinding
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
- <wsp:Policy>
- <sp:TransportToken>
- <wsp:Policy>
  <sp:HttpsToken RequireClientCertificate="false" />
  </wsp:Policy>
  </sp:TransportToken>
- <sp:AlgorithmSuite>
- <wsp:Policy>
  <sp:Basic256 />
  </wsp:Policy>
  </sp:AlgorithmSuite>
- <sp:Layout>
```

```
- <wsp:Policy>
  <sp:Strict />
  </wsp:Policy>
  </sp:Layout>
  <sp:IncludeTimestamp />
  </wsp:Policy>
  </sp:TransportBinding>
- <sp:SignedSupportingTokens
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
- <wsp:Policy>
- <sp:UsernameToken
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/
AlwaysToRecipient">
- <wsp:Policy>
  <sp:WssUsernameToken10 />
  </wsp:Policy>
  </sp:UsernameToken>
  </wsp:Policy>
  </sp:SignedSupportingTokens>
- <sp:EndorsingSupportingTokens
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
- <wsp:Policy>
- <sp:X509Token
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/
AlwaysToRecipient">
- <wsp:Policy>
  <sp:RequireThumbprintReference />
  <sp:WssX509V3Token10 />
  </wsp:Policy>
  </sp:X509Token>
- <sp:SignedParts>
  <sp:Header Name="To" Namespace="http://www.w3.org/2005/08/addressing" />
  </sp:SignedParts>
  </wsp:Policy>
  </sp:EndorsingSupportingTokens>
- <sp:Wss11 xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
- <wsp:Policy>
  <sp:MustSupportRefKeyIdentifier />
  <sp:MustSupportRefIssuerSerial />
  <sp:MustSupportRefThumbprint />
  <sp:MustSupportRefEncryptedKey />
  </wsp:Policy>
  </sp:Wss11>
- <sp:Trust10 xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
- <wsp:Policy>
  <sp:MustSupportIssuedTokens />
```

```
  <sp:RequireClientEntropy />
  <sp:RequireServerEntropy />
  </wsp:Policy>
  </sp:Trust10>
  <wsaw:UsingAddressing />
  </wsp:All>
  </wsp:ExactlyOne>
  </wsp:Policy>
- <wsdl:types>
- <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.exony.com/schemas/2009/10/status"
xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/" >
- <xsd:element name="OnStatusChange">
- <xsd:complexType>
- <xsd:sequence>
  <xsd:element minOccurs="0" name="status" nillable="true"
type="tns:ResourceStatus" />
  </xsd:sequence>
  </xsd:complexType>
  </xsd:element>
- <xsd:complexType name="ResourceStatus">
- <xsd:sequence>
  <xs:element minOccurs="0" name="ContainerId" type="ser:guid" />
  <xsd:element name="Id" nillable="true" type="xsd:string" />
  <xsd:element name="Status" nillable="true" type="xsd:string" />
  <xsd:element name="Type" nillable="true" type="xsd:string" />
  </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="ResourceStatus" nillable="true" type="tns:ResourceStatus" />
  </xsd:schema>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/">
  <xs:element name="anyType" nillable="true" type="xs:anyType" />
  <xs:element name="anyURI" nillable="true" type="xs:anyURI" />
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary" />
  <xs:element name="boolean" nillable="true" type="xs:boolean" />
  <xs:element name="byte" nillable="true" type="xs:byte" />
  <xs:element name="dateTime" nillable="true" type="xs:dateTime" />
  <xs:element name="decimal" nillable="true" type="xs:decimal" />
  <xs:element name="double" nillable="true" type="xs:double" />
  <xs:element name="float" nillable="true" type="xs:float" />
  <xs:element name="int" nillable="true" type="xs:int" />
  <xs:element name="long" nillable="true" type="xs:long" />
  <xs:element name="QName" nillable="true" type="xs:QName" />
```

```
  <xs:element name="short" nillable="true" type="xs:short" />
  <xs:element name="string" nillable="true" type="xs:string" />
  <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte" />
  <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt" />
  <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong" />
  <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort" />
  <xs:element name="char" nillable="true" type="tns:char" />
- <xs:simpleType name="char">
  <xs:restriction base="xs:int" />
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns:duration" />
- <xs:simpleType name="duration">
- <xs:restriction base="xs:duration">
  <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?" />
  <xs:minInclusive value="-P10675199DT2H48M5.4775808S" />
  <xs:maxInclusive value="P10675199DT2H48M5.4775807S" />
  </xs:restriction>
  </xs:simpleType>
  <xs:element name="guid" nillable="true" type="tns:guid" />
- <xs:simpleType name="guid">
- <xs:restriction base="xs:string">
  <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-
fA-F]{12}" />
  </xs:restriction>
  </xs:simpleType>
  <xs:attribute name="FactoryType" type="xs:QName" />
  <xs:attribute name="Id" type="xs:ID" />
  <xs:attribute name="Ref" type="xs:IDREF" />
  </xs:schema>
  </wsdl:types>
- <wsdl:message name="IStatus_OnStatusChange_InputMessage">
  <wsdl:part name="parameters" element="tns:OnStatusChange" />
  </wsdl:message>
- <wsdl:portType name="IStatus">
- <wsdl:operation name="OnStatusChange">
  <wsdl:input
wsaw:Action="http://www.exony.com/schemas/2009/10/status/onstatuschange"
message="tns:IStatus_OnStatusChange_InputMessage" />
  </wsdl:operation>
  </wsdl:portType>
- <wsdl:binding name="TrustedWebClientBinding_IStatus" type="tns:IStatus">
  <wsp:PolicyReference URI="#TrustedWebClientBinding_IStatus_policy" />
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="OnStatusChange">
```

```
  <soap12:operation
soapAction="http://www.exony.com/schemas/2009/10/status/onstatuschange"
style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
  </wsdl:input>
  </wsdl:operation>
  </wsdl:binding>
- <wsdl:service name="SinkStatus">
- <wsdl:port name="TrustedWebClientBinding_IStatus"
binding="tns:TrustedWebClientBinding_IStatus">
  <soap12:address location="https://localhost:33333/Sink" />
- <wsa10:EndpointReference>
  <wsa10:Address>https://localhost:33333/Sink</wsa10:Address>
  </wsa10:EndpointReference>
  </wsdl:port>
  </wsdl:service>
  </wsdl:definitions>
```