Cisco Unifieid Contact Center Management Portal Web Services Reference Guide, Release 12.6(1)

For Unified Contact Center Enterprise

First Published: May, 2021

Americas 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: 408 527-0883

Cisco Unifieid Contact Center Management Portal Web Services Reference Guide: For Unified Contact Center Enterprise. May 11, 2021

Contents

Preface

About This Guide

Cisco Unified Contact Center Management Portal Web Services Reference Guide describes how to manage and maintain user security.

Who Should Read This Document

This document is for administrators responsible for the commissioning and ongoing maintenance of Unified CCMP. All users responsible for managing security should have access to this document and to the details of their specific Unified CCMP system configuration. The reader should be familiar with the Unified CCMP web interface, as described in the Unified CCMP User Guide, and have a general understanding of contact center operations.

Product Naming Conventions

In this release, the product names defined in the table below have changed. The New Name (long version) is reserved for the first instance of that product name and in all headings. The New Name (short version) is used for subsequent instances of the product name.

Note: This document uses the naming conventions provided in each GUI, which means that in some cases the old product name is in use.

Old Product Name	New Name (long version)	New Name (short version)
Cisco IPCC Enterprise Edition	Cisco Unified Contact Center Enterprise	Unified CCE
Cisco IPCC Hosted Edition	Cisco Unified Contact Center Hosted	Unified CCH
Cisco Intelligent Contact Management (ICM) Enterprise Edition	Cisco Unified Intelligent Contact Management (ICM) Enterprise	Unified ICM
Cisco Intelligent Contact Management (ICM) Hosted Edition	Cisco Unified Intelligent Contact Management (ICM) Hosted	
Cisco CallManager/Cisco Unified CallManager	Cisco Unified Communications Manager	Unified CM
Cisco CallManager/Cisco Unified CallManager	Cisco Unified Communications Manager	Unified CM

Related Documentation

Documentation for Cisco Unified ICM/Contact Center Enterprise & Hosted, as well as related documentation, is accessible from Cisco.com at: https://www.cisco.com/cisco/web/psa/default.html.

Related documentation includes the documentation sets for:

- Cisco CTI Object Server (CTIOS), Cisco Agent Desktop (CAD)
- Cisco Agent Desktop - Browser Edition (CAD-BE)
- Cisco Unified Contact Center Domain Manager
- Cisco Unified Contact Center Management Portal
- Cisco Unified Customer Voice Portal (CVP)
- Cisco Unified IP IVR, Cisco Unified Intelligence Center
- Cisco Support Tools

Documentation for these Cisco Unified Contact Center products is accessible from:

- https://www.cisco.com/cisco/web/psa/default.html.

Documentation for Cisco Unified Communications Manager is accessible from:

- https://www.cisco.com/cisco/web/psa/default.html.

Technical Support documentation and tools are accessible from:

- https://www.cisco.com/en/US/support/index.html.

The Product Alert tool is accessible from (sign in required):

- https://www.cisco.com/cgi-bin/Support/FieldNoticeTool/field-notice.

For information on the Cisco software support methodology, refer to *Software Release and Support Methodology: ICM/IPCC,* available from (sign in required):

- https://www.cisco.com/en/US/partner/products/sw/custcosw/ps1844/prod_bulletins_list.html.

For a detailed list of language localizations, refer to the *Cisco Unified ICM/Contact Center Product and System Localization Matrix,* available from:

- https://www.cisco.com/en/US/products/sw/custcosw/ps1001/prod_technical_reference_list.html.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

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 https://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.

Document Conventions

This guide uses the following typographical conventions.

Convention	Indicates
Italic	Emphasis, or the title of a published document.
Bold	An item in the user interface, such as a window, button, or tab.
`Monospace`	A file name or command.
Script	A variable, which is a placeholder for user-specific text provided by the user. Or, text that must be typed by the user.

Document conventions

1 Getting Started

About Unified Contact Center Management Portal

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

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

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

For a technical overview of Unified CCMP, including architecture, design concepts, and performance tips, see the *Unified CCMP User Guide*.

About Unified CCMP Web Service Protocols

The Unified CCMP Web Services supports Representational state transfer, or REST protocol.

REST

The provisioning resources are organized into collections to allow clients and the Unified CCMP 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 CCMP 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
```

Additionally, Unified CCMP supports a number of addressing modes which can be seen in the table below.

	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

Security

The Unified CCMP Web Services use Unified CCMP'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 CCMP Web Services are secured using WS-Security user name tokens. You must pass a valid Unified CCMP user name and password to access the service methods.

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.

For example, this might look like:

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

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

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 "About Securing the Subscriptions Web Service" on page 214.

Common Data Types and Remote Resource Types

Remote Resource Types

About Remote Resource Types

These types correspond directly to the remote resources in the Contact Center environment in which Unified CCMP operates. These types can be used with the Resource Management Web Service APIs (see Chapter "Resource Management Web Service") to allow client utilities and a mid-tier interface to perform adds, updates and deletes. Some of these types can be used by the Hierarchy Management Web Service APIs (see Chapter "Hierarchy Management Web Service") for hierarchy node association.

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
Device Profile	IT_DEVICE_PROFILE	device-profile
Dialed Number	IT_DIALED_NUMBER	dialed-number
Directory Number	IT_DIRECTORY_NUMBER	directory-number
Enterprise Skill Group	IT_ENTERPRISE_SKILLGROUP	enterprise-skill-group
Expanded Call Variable	IT_EXPANDED_CALL_VARIABLE	expanded-call-variable
IP Endpoint	IT_IP_ENDPOINT	ip-endpoint
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
Precision Attribute	IT_PRECISION_ATTRIBUTE	precision-attribute
Precision Queue	IT_PRECISION_QUEUE	precision-queue

Resource Type	Internal Name	REST Parameter
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

Non Provisionable Remote Resource Types

These remote resource types are supported by the Resource Management Web Services for searching only. Where indicated, they are also supported by the Hierarchy Management Web Service.

Resource Type	Internal Name
Announcement	IT_ANNOUNCEMENT
Application Gateway	IT_APPLICATION_GATEWAY
Application Instance	IT_APPLICATION_INSTANCE
Bucket Interval	IT_BUCKET_INTERVAL
Call Manager Group	IT_CALL_MANAGER_GROUP
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 Pool	IT_DEVICE_POOL
Device Target	IT_DEVICE_TARGET
Dialer	IT_DIALER
Dial Number Plan	IT_DIAL_NUMBER_PLAN
Enterprise Route	IT_ENTERPRISE_ROUTE
Enterprise Service	IT_ENTERPRISE_SERVICE

Resource Type	Internal Name
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
IP Endpoint Button Template	IT_IP_ENDPOINT_BUTTON_TEMPLATE
IP Endpoint Model	IT_IP_ENDPOINT_MODEL
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
Route Partition	IT_ROUTE_PARTITION
Routing Client	IT_ROUTING_CLIENT
Routing Script	IT_ROUTING_SCRIPT

Resource Type	Internal Name
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

System Resource Types

About System Resource Types

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

Editable System Resource Types

These system resource types are fully supported by the Resource Management Web Services. They are not supported by the Hierarchy Management Web Service.

Resource Type	Internal Name	REST Parameter
Category	IT_CATEGORY	category
Code Group	IT_CODE_GROUP	code-group
Code	IT_CODE	code
Document	IT_DOCUMENT	document
Folder	IT_FOLDER	folder

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

Read-Only System Resource Types

These system resource types are supported by the Resource Management Web Services for searching only. They can also be used by the Hierarchy Management Web Services where indicated.

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	T_CLUSTER_RESOURCE_TYPE
Cluster Resource Type Group	IT_CLUSTER_RESOURCE_TYPE_GROUP
Cluster Server	IT_CLUSTER_SERVER
Code Constraint Type	IT_CODE_CONTRAINT_TYPE
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

Resource Type	Internal Name
Custom 8	IT_CUSTOM8
Custom 9	IT_CUSTOM9
Fileshare	IT_FILESHARE
Email	IT_EMAIL
Form Template	IT_FORM_TEMPLATE
Gadget Provider	IT_GADGET_PROVIDER
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

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.

- 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. The type-2 changes in items caused by the Hierarchy Management Web Service are hidden from the Resource Management Web Service.

Important: **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 an SCD event, a remote resource may have the following values:

Identifier	Surrogate	EffectiveFrom	EffectiveTo	Latest
1000	1000	01/01/2012	02/02/2012	1
1000	1001	01/01/2012	06/062079	

The surrogate key field is used to track the change.

3 Resource Management Web Service

Resource Management Web Service

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.

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 man-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.
- 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 lifecycle can be tracked. Type-2 SCD lifecycle 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 or existing clients can be changed.
  - Versioning can be performed at the resource of 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 (such as a request to create a Folder) use a synchronous request and a synchronous response.
  - Provisioning transactions for remote resources (such as a request to create an Agent) use a synchronous request and an asynchronous response for maximum scaling. Remote resources will go through a Pending state before they are successfully provisioned and enter the Ready state.
- 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.

Item Types

The Unified CCMP 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 "Provisionable Remote Resource Types" on page 70)

- Non-provisionable remote resource types (see "Non-Provisionable Remote Resource Types" on page 126.
- Provisionable system resource types (see "Editable System Resource Types" on page 62.
- Non-provisionable system resource types (see "Read-Only System Resource Types" on page 30.
- Membership types (see "Resource Management Web Service Member Types" on page 126.

Pkey Map Classes

The Unified CCMP 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 CCMP 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 CCMP 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 CCMP membership with the corresponding agent to the agent team relationship on the remote equipment.

Important: **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.**

Methods

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

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

- `Describe()`
- `Audit()`
- `Upload()`
- `Download()`
- `Deploy()`

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.

About Enterprise-Level Caching

Due to the distributed environment in which the Unified CCMP 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.

The table below shoes which APIs provide direct database-level access and which APIs rely on cached data from the enterprise-level cache.

API	Direct Database Access	Enterprise Level Cache
Create()	Yes	
Update()	Yes	
Delete()	Yes	
Save()	Yes	
Move()	Yes	
Retrieve()		Yes
Search()		Yes
Describe()		Yes
Audit()		Yes

Note that notifications sent from the Unified CCMP application server also rely on updates from the enterprise-level cache, and as such, will only be sent when this cache is updated.

Resource Management Web Service Specification

REST

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

Errors

The table below details various possible errors, their error codes, and the identifiers for each error. Note that not every error code is detailed here.

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.
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	ConcurrencyConflictException	The resource to be updated has already been updated by another process, that is, its ChangeStamp field is too old.
101	ServiceNotReadyException	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	InvalidConditionException	Search request error: the system could not parse the supplied search term condition, for example, specifying type:Aggent rather than type:Agent.
502	FieldConversionException	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	EntityValidationException	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	ExonyApplicationException	Last try application exception.

Error Code	Identifier	Description
50150	NoMembershipEffectivenessOverlap	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	RequiredFieldNotSpecified	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	RegularExpressionValidationFailed	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.
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.
1000001	AgentMultiplePersonality	Future use.
1000002	PeripheralRequired	Future use.
1000003	RoutingClientRequired	Future use.
1000004	IPEndpointMultipleLines	A type of EntityValidationException error. Indicates that an attempt has been made to provision more than one line on a device.
1000005	UnknownClusterResource	A type of ExonyApplicationException error. Indicates that the supplied cluster resource type is not a supported type.
1000006	AgentsInTeamCapacityExceeded	Future use.
1000007	SkillgroupIPTA	A type of EntityValidationException error. Indicates that for voice, skill groups must let the system pick the agent.
100039	CannotFindDefaultRoutingClient	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	CannotFindDefaultPeripheral	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	CannotFindDefaultOutboundRoutingClient	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.

Error Code	Identifier	Description
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	LoginNameChangeNotSupported	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.
100074	NoEquipmentMappedToTenant	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 *Unified CCMP Administration Guide*) 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	EnterpriseNameAlreadyExistsException	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	LoginNameAlreadyExistsException	The system has detected that a User.LoginName already exists. The LoginName must be globally unique across the installation.
50222	InvalidFromDateGreaterThanTo	The ‘effective from’ date is newer than the ‘effective to’ date.
50176	FieldValueLengthInvalid	The description provided exceeds the 255 character limit.
50146	InvalidEffectiveToSetToPast	The ‘effective to’ date has already passed.
100066	PersonUnavailable	The person being used to create an agent has already been associated with another person.
55513	DomainLoginNamedoesNotExist	A supervisor agent has been created with an invalid domain login name.
50316	ValueNotEqualToCodeConstraintType	A code has been created with an invalid code group constraint type.
55511	UpdateToFieldNotAllowed	A code group constraint type has been changed.
100052	UnknownCodeConstraintType	A code group has been created using an unrecognized constraint type.
60041	FolderNotEmpty	The folder being deleted contains resources.

Restrictions

Array Limits

The Unified CCMP 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:

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
Retrieve()	255 Resource Keys	255 Resources
Search()	None	1000 Resources

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.s

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:

Resource	Field	Regular Expression	Length
Various	Internal Field	`^[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

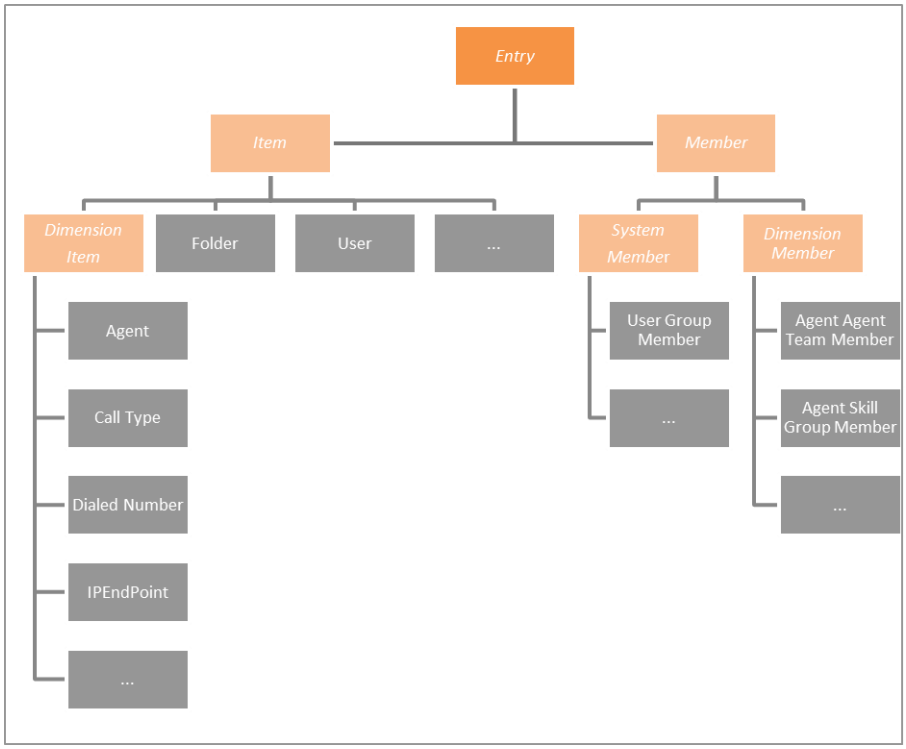
Resource	Field	Regular Expression	Length
Dialed Number	Dialed Number String	`.*`	32
Directory Number	Number Pattern	`^[X0-9\?\[\]\+\-\*\^#!]+$`	12
IP Endpoint	Description	`^[\]a-zA-Z0-9\s#\$'\(\)\*\+,\./:;=\?@\[\^_`\{|\}~\-]*$`	32
Various	MAC Address	`^([0-9a-fA-F][0-9a-fA-F]){5}([0-9a-fA-F][0-9a-fA-F])$`	32
Person	First/Last Name	`^[a-zA-Z0-9][a-zA-Z0-9_\-\.\s]*$`	27
Person	Peripheral Login Name	`^[a-zA-Z0-9][a-zA-Z0-9_\.]*$`	32
Expanded Call Variable/User Variable	Name/Internal Name	`^[a-zA-Z0-9][a-zA-Z0-9_\-\.\s]*$`	27

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.

The Resource Hierarchy

The item and member types are represented by the resource class hierarchy.



The Resource Hierarchy displayed.

The top-level abstract class, `Entry`, provides a simple-type definition for an entity. Requests to retriever 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 2018 and be part of a different Agent Team from June 2018 to February 2019.

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

Common Fields and Classes

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:

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
StorageType	NameValuePair StorageType	The storage type for this name-value pair. One of: Standard: Used by Unified CCMP 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)

Important: **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 REST protocols.**

Status

Remote Resources

The Status field is used to report the status of a remote resource. 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 CCMP 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.
Delete	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 CCMP 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.

Note that ISE-enabled users are different and follow the state transitions outlined in “Remote Resource State Transitions.” This is due to additional state transitions applying when the ISE-enabled option is set or cleared, and when certain other changes are made to an ISE-enabled user.

Resource Hierarchy Classes

The resource hierarchy is composed of several different classes.

- Entry: The Entry class represents a primary Unified CCMP entity or membership.

- Properties: The exposed properties on the Entry class are:

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.	No
Type	String	A string identifying the item type. Consists of one of the following: - A remote resource type - A system resource type - A member type - A pkey type - A pkey member type	Yes
EffectiveFrom	DateTime	The UTC date that the entity will become effective or has become effective in the contact center environment.	No
EffectiveTo	DateTime	The UTC date that the entity will expire or has expired and been removed or deactivated in the contact center environment.	No
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.	No
Fields	NameValuePair[]	A collection of the fields that have been set or may be set on the current resource.	Yes
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 and the resource is mapped to more than one equipment instance. - Type is one of the resource membership types listed in "Member Types and Identifiers" on page 126 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 . For more information about the use of this field when retrieving resources and memberships, see "Retrieving Items with Pkey Maps" on page 155.	No
Custom	Int	User-defined custom data.	No
CreatedById	Grid (n-v pair)	Read-only id of the user that created this item, returned on retrieve or search responses.	n/a

Element Name	Data Type	Description	Required?
ModifiedById	Grid (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
Delated	Boo1 (n-v pair)	Read-only flag indicating whether the item has been deleted.	n/a
Hidden	Boo1 (n-v pair)	Read-only flag indicating whether the item has been hidden.	n/a
Latest	Boo1 (n-v pair)	Read-only flag indicating whether this resource is the latest in a type-2 sequence.	n/a
System	Boo1 (n-v pair)	Read-only flag indicating whether the item is owned by the system.	n/a

Important: **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 REST protocols.**

Item

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

Properties

The exposed Properties in the Item class are as follows:

Element Name	Data Type	Description	Required?
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

Dimension Item

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

The exposed properties on the `DimensionItem` class inherit from `Item` and are:

Element Name	Data Type	Description	Required?
HierarchyParentUrn	Int(n-v pair)	Read-only. The type-1 Hierarchy for this dimension item.Default: -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
Description	String(n-v pair)	The description of this instance of the resource.	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

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 listed below. They inherit from `Entry`.

Element Name	Data Type	Description	Required?
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 parent resource for the membership.	Yes

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.

The exposed properties on the `DimensionItem` class are:

Element Name	Data Type	Description	Required?
LatestParentItemUrn	Int(n-v pair)	This is the id of the resource which is the parent resource for the membership.	Yes
LatestChildItemUrn	Int(n-v pair)	This is the id of the resource which is the parent resource for the membership.	Yes

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:

Property	Data Type	Description	Required?
LatestParentItemUrn	Int(n-v pair)	This is the id of the resource which is the parent resource for the membership.	Yes
LatestChildItemUrn	Int(n-v pair)	This is the id of the resource which is the parent resource for the membership.	Yes

Other Common Classes

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

Resource Meta

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

Properties

The exposed properties of the `ResourceMeta` class are listed below.

Property	Data Type	Description
Name	String	The name of the field that may be set/read on the entity
Data Type	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. ▸ Equipment. The field contains equipment cluster reference.
HelpText	String	A description of the use of this field.

Resource Field Meta

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

Properties

The exposed properties of the `ResourceMeta` class are listed below.

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 "Remote Resource Types" on page 26. ▸ a system resource type from "System Resource Types" on page 29. ▸ a member type from "Member Types and Identifiers" on page 126. ▸ a pkey type from "Provisionable Remote Resource Types" on page 70 or section "Non-Provisionable Remote Resource Types" on page 126 ▸ a pkey member type from "System Resource Pkey Map Types" on page 58.
CanCreate	Bool	An instance of this resource may be created in the Unified CCMP database.
CanEdit	Bool	An instance of this resource may be edited.
CanDelete	Bool	An instance of this resource may be deleted from the Unified CCMP 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 CCMP 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.

Resource Key

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

Properties

The exposed properties of the `ResourceKey` class are listed below.

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.

Resource Audit

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

Properties

The exposed properties of the `ResourceAudit` class are listed below.

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 an Unified CCMP 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.
Resource Name	String	The name of the resource that is the subject of the audit.

Detail Fields

Element Name	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.

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 of the `RequestResult` class are listed below.

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.
Errors	String[]	The error messages returned as a result of changing this item. If no errors occurred, this is an empty array.

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 of the `ResourceFaultDetail` class are listed below.

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
Name	String	One of the error codes listed in section "Errors".

Additional Data

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

Properties

The exposed properties of the `AdditionalData` class are listed below.

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

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 of the `PageInfo` class are listed below.

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.

Resource Audit Results

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

Properties

The exposed properties of the `ResourceAuditResults` class are listed below.

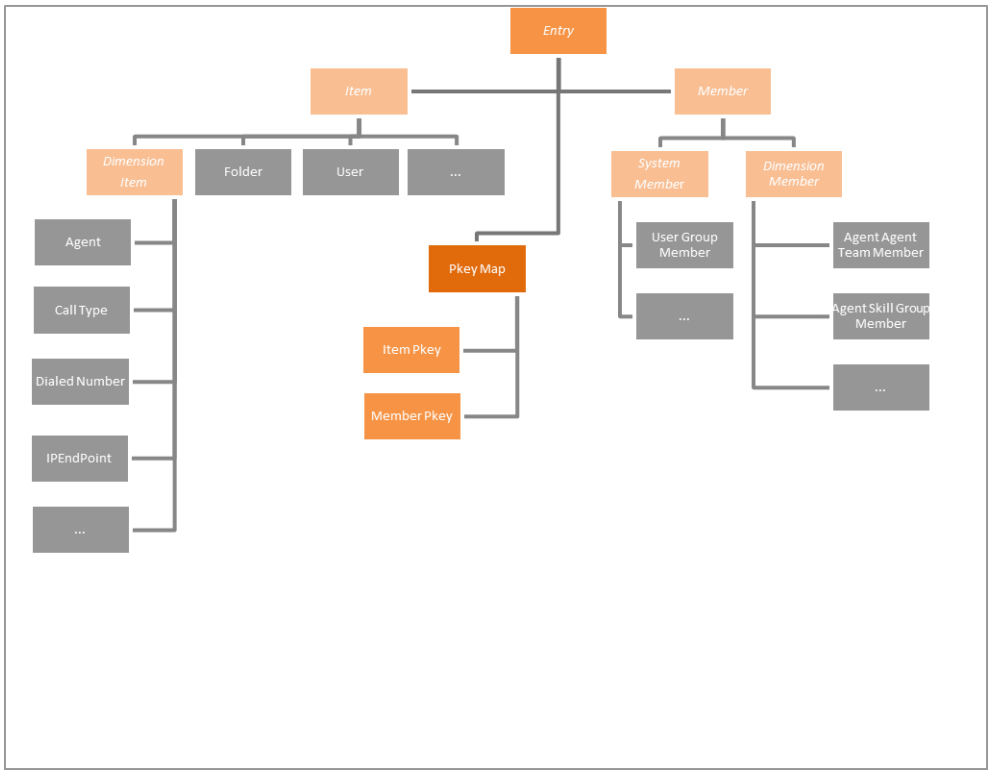
Property	Data Type	Description
PageInfo	PageInfo	Information about the collection of audit results being returned. For more information, see "Page Info" on page 53.
Audits	ResourceAudit[]	A collection of ResourceAudit objects containing audit data for the resource id passed. For more information, see "Resource Audit" on page 51.

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.

Note that 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.

Resource Hierarchy Including Pkey Maps



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
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 Pkeu	PK_DEPARTMENT_PKEY	department-pkey
Device Profile Pkey	PK_DEVICE_PROFILE_PKEY	device-profile-pkey
Dialed Number Pkey	PK_DIALED_NUMBER_PKEY	dialed-number-pkey
Directory Number Pkey	PK_DIRECTORY_NUMBER_PKEY	directory-number-pkey

Resource Pkey Map Type	Internal Name	REST Parameter
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
IP Endpoint Pkey	PK_IP_ENDPOINT_PKEY	ip-endpoint-pkey
Label Pkey	PK_LABEL_PKEY	label-pkey
Network VRU Script Key	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

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
Call Manager Group Pkey	PK_CALL_MANAGER_GROUP_PKEY
Call Source Pkey	PK_CALL_SOURCE_PKEY

Resource Pkey Map Type	Internal Name
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 Pool Pkey	PK_DEVICE_POOL_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
IP Endpoint Button Template Pkey	PK_IP_ENDPOINT_BUTTON_TEMPLATE_PKEY
IP Endpoint Model Pkey	PK_IP_ENDPOINT_MODEL_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

System Resource Pkey Map Types

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

Resource Pkey Map Type	Internal Name	REST Parameter
Code Pkey	PK_CODE_PKEY	code-pkey
Code Group	PK_CODE_GROUP_PKEY	code-group-pkey

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
Code Code Group Member Pkey	PK_CODE_CODE_GROUP_MEMBER_PKEY	code-code-group-member-pkey
Device Profile Directory Number Member Pkey	PK_DEVICE_PROFILE_DIRECTORY_NUMBER_MEMBER_PKEY	device-profile-directory_number-member-pkey
Dialed Number Call Type Member Pkey	PK_DIALED_NUMBER_CALL_TYPE_MEMBER_PKEY	dialed-number-call-type-member-pkey
IP Endpoint Directory Number Member Pkey	PK_IP_ENDPOINT_DIRECTORY_NUMBER_MEMBER_PKEY	ip-endpoint-directory-number-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

Member Pkey Map Type	Internal Name	REST Parameter
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

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:

Element Name	Data Type	Description	Required?
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
ClusterResourceType	String(n-v pair)	The type of cluster resource, eg. CRT_CISCO_CICM.	No

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 CCMP or imported into Unified CCMP.

Properties

The exposed properties on the `ItemPkey` class are:

Inherits From	PkeyMap where Type is a pkey type

Element Name	Data Type	Description	Required?
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

Member Pkey Class

The `MemberPkey` class applies to memberships 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 CCMP or imported into Unified CCMP.

Properties

The exposed properties of the `MemberPkey` class are:

Inherits From	PkeyMap where Type is a member pkey type

Element Name	Data Type	Description	Required?
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

More Information

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

- Each of the provisionable resource types described in "Provisionable Remote Resource Types" on page 70 includes information about the fields that may be located on pkey items.
- Each of the APIs described in "Resource Management Web Service APIs" on page 146 explains how to use pkey maps with that API.

Editable System Resource Types

Folder Resource Item

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

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.

Inherits From	`Item`

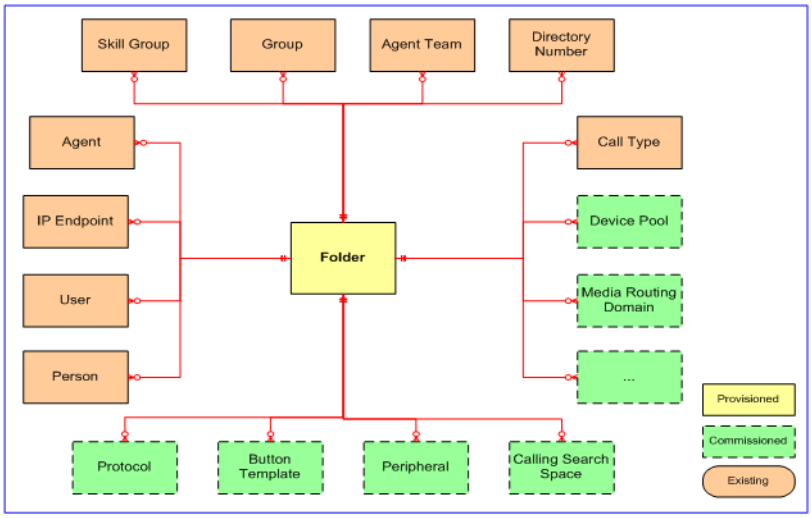
Element Name	Data Type	Description	Required ?
Path	String (n-v pair)	The path defining the location of this folder. This is a read-only field set by Unified CCMP on Retrieve or Search operations only.	Yes
ParentId	Guid (n-v pair)	The identifier of the folder that is the immediate parent of this folder. This is a read-only field set by Unified CCMP 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 CCMP 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. This is a read-only field set by the application on Retrieve or Search operations 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 CCMP on Retrieve or Search operations only.	Yes

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

Associated Name 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.

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



Resource folder relationships

REST Protocol

The following information is a summary for the Folder REST API.

Base URL	https://<server>:8085/resourcemanagement/rest/resources
IO Format	JSON

Operation	Mode	Method	Status	URL
Audit	Sync.	GET	200 OK	[Base URL]/folders/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>â?¦
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>

Operation	Mode	Method	Status	URL
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>

Exception	See "Errors" on page 37
Example	POST https://Web01:8085/resourcemanagement/rest/resources/folders

Group Resource Item

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

Fields

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

Inherits From	`Item`

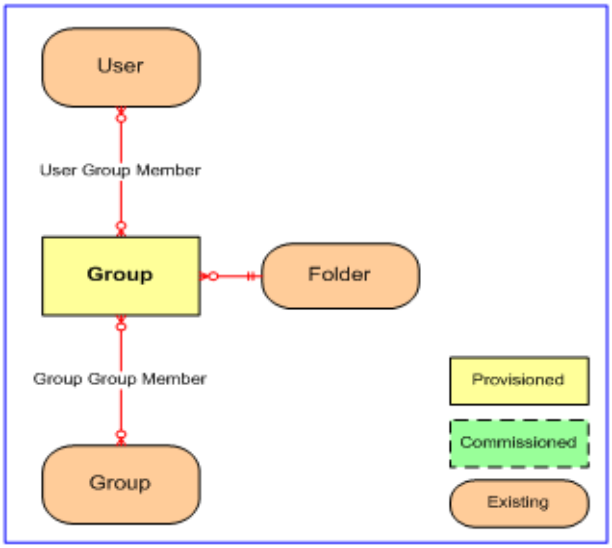
Element Name	Data Type	Description	Required?
Enabled	Bool (n-v pair)	Flag indicating whether the group is enabled. Used to disable a security group.	No
Hidden	Bool (n-v pair)	Flag indicating whether the group is hidden. Used for system security groups that are not visible to tenants.	No

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

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.

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



Group relationships

REST Protocol

The following is a summary for the Code Group REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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
Delete	Sync.	DELETE	202 Accepted	[Base URL]/groups/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/group

Operation	Mode	Method	Status	URL
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]/code-groups/<id>,<id>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/code-groups				

User Resource Item

The `User` resource is the login record for an Unified CCMP 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 CCMP 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.

Fields

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

Element Name	Data Type	Description	Required?
Inherits From `Item`			
LoginName	String(70) (n-v pair)	The login name for an Unified CCMP 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 user name 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.	Yes if local login is enabled
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

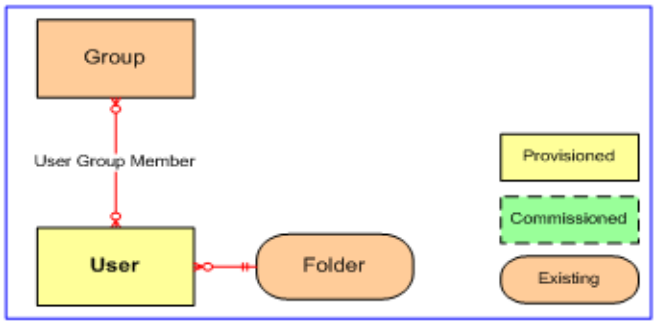
Element Name	Data Type	Description	Required?
PassPhraseChangeEnabled	Bool (n-v)	A flag indicating whether the user can change their passphrase or not. A value of true indicates that the user can change their passphrase. Default is true.	No
CreateHomeFolder	Bool (n-v pair)	Used on create only, to indicate 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
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 "Valid Time Zones" on page 228 for the list of valid time zones.	No
LocalLoginEnabled	Bool	Indicates whether this user can login to the system using Local Login.	No

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

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.

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



User relationships

REST Protocol

The following is a summary for the Folder REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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

Operation	Mode	Method	Status	URL
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/users/<id>,<id>
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]/code-users/<id>,<id>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/users				

Provisionable Remote Resource Types

Agent Resource Item

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.

Fields

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

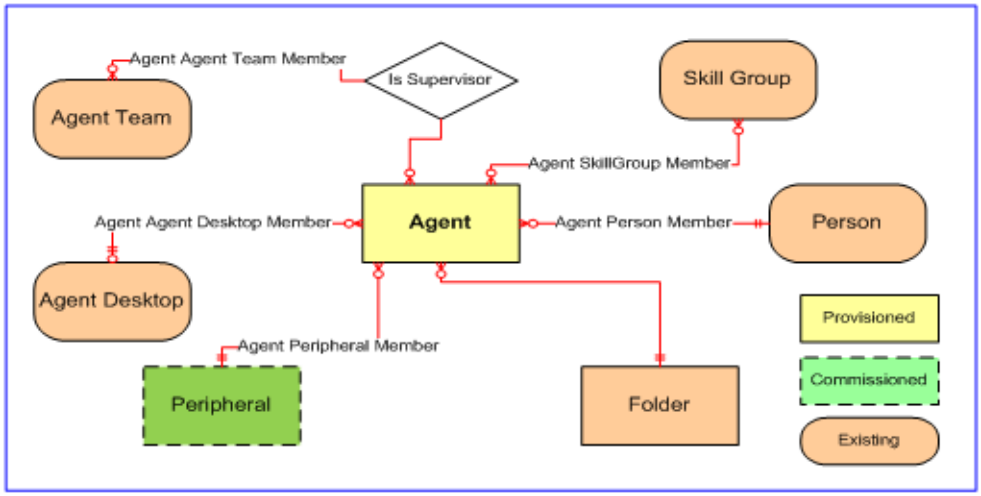
Element Name	Data Type	Description	Required?	Pkey?
Inhabits From: `DimensionItem`				
PeripheralUrn	String(10) (n-v pair)	The peripheral or equipment identifier on which the agent will be located. If -1 is supplied then Unified CCMP will pick the Peripheral automatically. Note: cannot be updated after 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

Element Name	Data Type	Description	Required?	Pkey?
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
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.	Yes	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 CCMP background processes. Note: used for Unified CCE parent/child configurations only.	No	Yes
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

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.

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



Agent relationships

REST Protocol

The following is a summary for the Agent REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources

- **ID Format:** JSON

Operation	Mode	Method	Status	URL
Audit	Sync.	GET	200 OK	[Base URL]/agents/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>â?¦
Create	Async.	POST	202 Accepted	[Base URL]/agents
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/agents				

Agent Desktop Resource Item

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.

Fields

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

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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

Element Name	Data Type	Description	Required?	Pkey?
AvailableAfterOutgoing	Bool(n-v pair)	A flag indicating whether to consider an agent to be available after handling an outbound call.	No	Yes
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; 3: Required with Wrap up Data	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
SupervisorAssistCallMethod	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; None.	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; None.	No	Yes

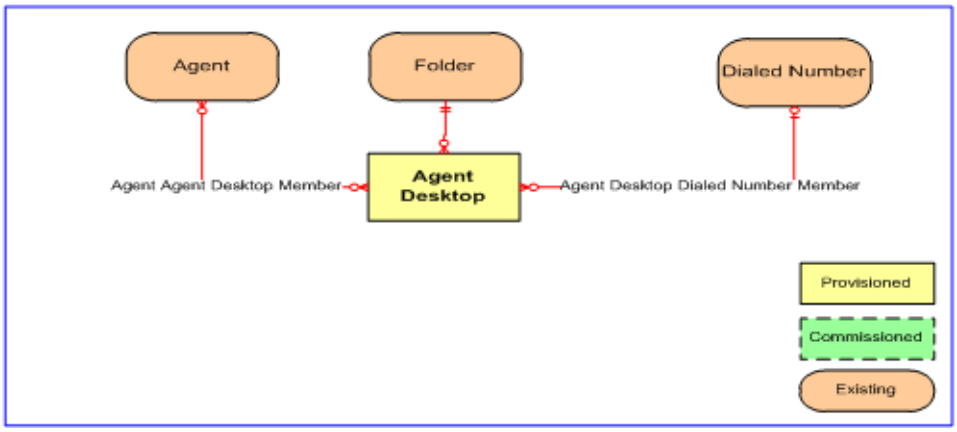
Element Name	Data Type	Description	Required?	Pkey?
AutoRecordOnEmergency	Bool(n-v pair)	A flag indicating whether to automatically record or not to record when an emergency call request is started.	No	Yes
AgentToAgentCallsAllowed	Bool(n-v pair)	A flag indicating whether calls to other agents are allowed.	No	Yes
OutboundAccessInternational	Bool(n-v pair)	A flag indicating whether the agent can initiate international calls.	No	Yes
OutboundAccessOperatorAssisted	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
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
SilentMonitorWarningMessage	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
SilentMonitorAudibleIndication	Bool(n-v pair)	A flag indicating whether an audio click will sound when silent monitoring is started.	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

Element Name	Data Type	Description	Required?	Pkey?
RemoteLoginWithoutDesktop	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

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.

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



Agent desktop relationships

REST Protocol

The following table is a summary for the Agent REST API.

- **Base URL:** `https://<server>:8085/resourcemanagement/rest/resources`

ID Format: 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
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/agent-desktops				

Agent Team Resource Item

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. 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.

Fields

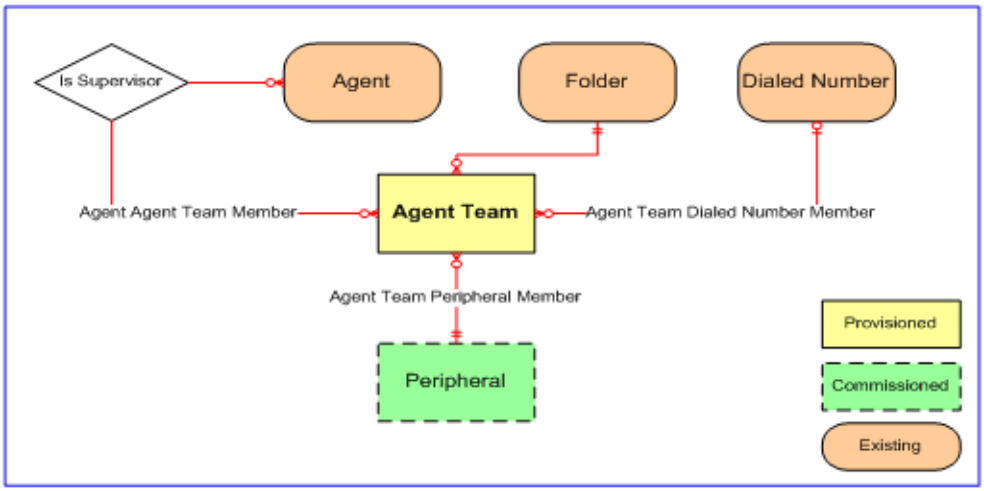
The following fields can be read or set for Agent Team resources.

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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 CCMP 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

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.

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



Agent team relationships

REST Protocol

The following table is a summary for the Agent Team REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/agent-teams				

Call Type Resource Item

The `CallType` resource represents a classifier applied to a specific class of calls. This enables 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.

Fields

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

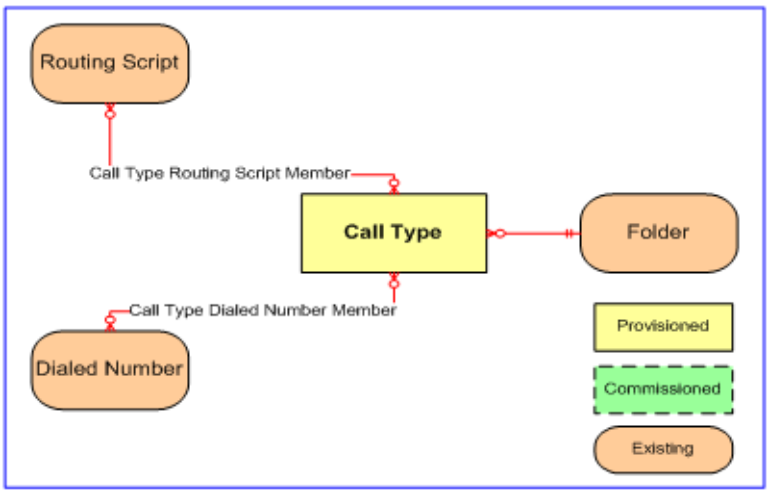
Element Name	Data Type	Description	Required?	Pkey?
ServiceLevelThreshold	Int(n-v pair)	The service level threshold for the Call Type. Possible values are: 0: None (default) 1: Use global settings 2 or greater: Threshold value in seconds	No	Yes
ServiceLevelType	Short(n-v pair)	This is the type that defines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

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.

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

r



Call type relationships

REST Protocol

The following tables are a summary for the Call Type REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/call-types				

Calling Search Space Resource Item

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

Fields

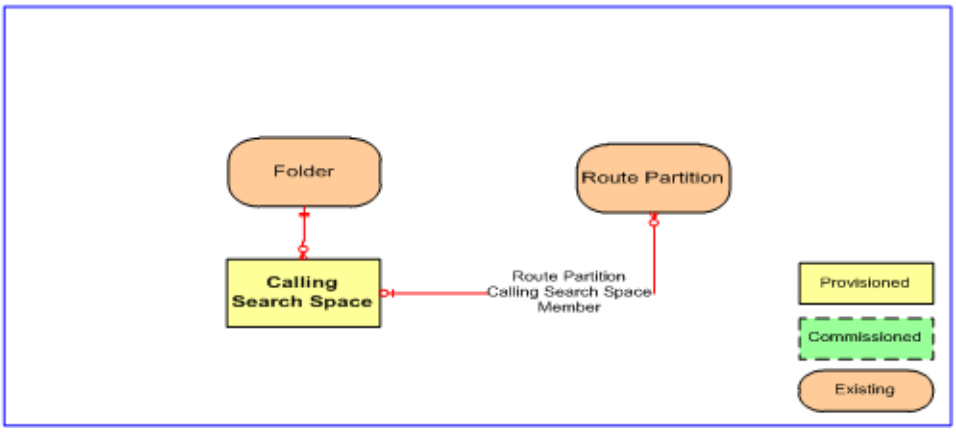
The following fields can be read or set for Calling Search Space resources.

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

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.

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



Calling Search Space relationships

REST Protocol

The following table is a summary for the Calling Search Space REST API.

- **Base URL:** `https://<server>:8085/resourcemanagement/rest/resources`

- **ID Format:** 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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/calling-search-spaces				

Department Resource Item

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

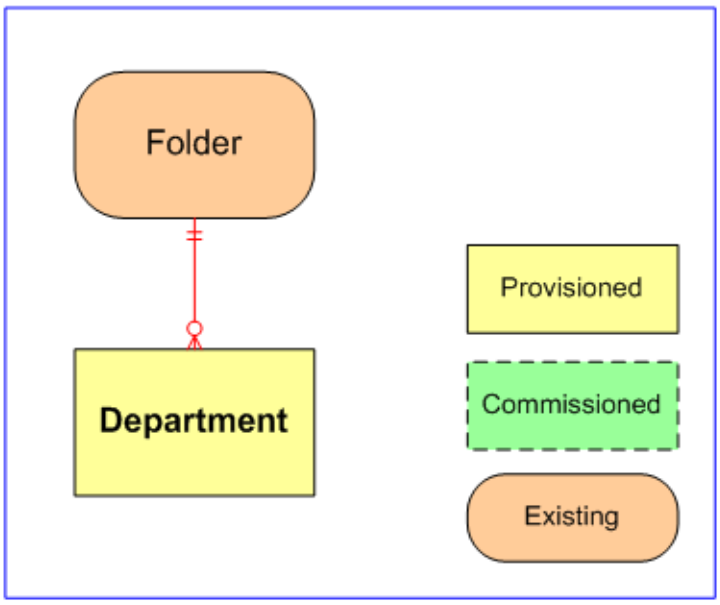
Fields

There are no further fields for this resource.

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.

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



Department relationships

REST Protocol

The following table is a summary for the Department REST API.

- **Base URL:** `https://<server>:8085/resourcemanagement/rest/resources`
- **ID Format:** 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

Operation	Mode	Method	Status	URL
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>
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/departments				

Device Profile Resource Item

The `DeviceProfile` represents a collection of fields that are associated with a particular device.

Fields

The following fields can be read or set for Device Profile resources:

Element Name	Data Type	Description	Required?	Pkey?
IPEndpointButtonTemplateUrn	String(n-v pair)	Parent resource that must be supplied during the create process. Note: this is only applicable to devices which support button templates.	Yes, if the device supports button templates, otherwise no.	No
ProductCodeBizUrn	Int(n-v pair)	Foreign key to a product code. Code type: CG_DEVICE_PRODUCT	Yes	No
ProtocolCodeBizUrn	Int(n-v pair)	Foreign key to a protocol code. Code type: CG_DEVICE_PROTOCOL	Yes	No
TemplateData	Xml(n-v pair)	The template attributes that will be used as a default for any IP Endpoint Creation	No	No

Associated Name Fields for Device Profiles

As well as the fields above, the following associated name fields can also be used to sort Device Profile 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.

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

REST Protocol

The following table is a summary for the Device Profile REST API.

- **Base URL:** `https://<server>:8085/resourcemanagement/rest/resources`

- **ID Format:** JSON

Operation	Mode	Method	Status	URL
Audit	Sync.	GET	200 OK	[Base URL]/device-profiles/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate> …
Create	Async.	POST	202 Accepted	[Base URL]/device-profiles
Delete	Async.	DELETE	202 Accepted	[Base URL]/device-profiles/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/device-profile
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/device-profiles/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/device-profiles
Search	Sync.	GET	200 OK	[Base URL]? queryString=type%3aIT_DEVICE_PROFILE&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/device-profiles/<id>,<id>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/device-profiles				

Dialed Number Resource Item

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.

Fields

The following fields can be read or set for Dialed Resource numbers:

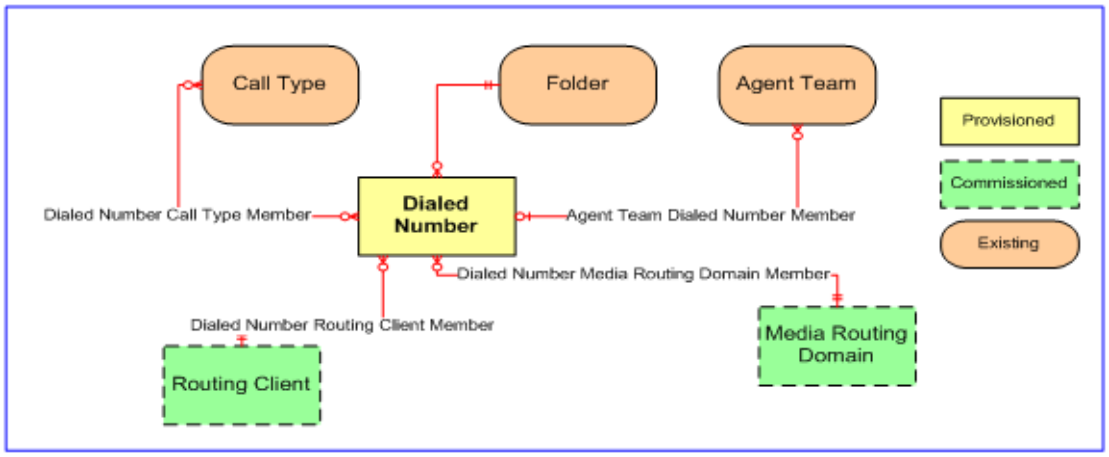
Element Name	Data Type	Description	Required?	Pkey?
Inhabits From `DimensionItem`				
RoutingClientUrn	String(10) (n-v pair)	The routing client that services this dialed number.	Yes	No
MediaRoutingDomainUrn	String(10) (n-v pair)	The media routing domain that services this dialed number.	Yes	No
DialedNumber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

Element Name	Data Type	Description	Required?	Pkey?
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

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

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



Dialed number relationships

REST Protocol

The following table is a summary for the Dialed Number REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources

- **ID Format**: 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
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/dialed-numbers				

Directory Number Resource Item

The `DirectoryNumber` resource represents an extension number or number pattern on a switch. This type is only created by Unified CCMP when Unified CCMP is the only provisioning tool. When other provisioning tools are used then Unified CCMP detects new instances of this type by reading the switch directly as part of the import process.

Fields

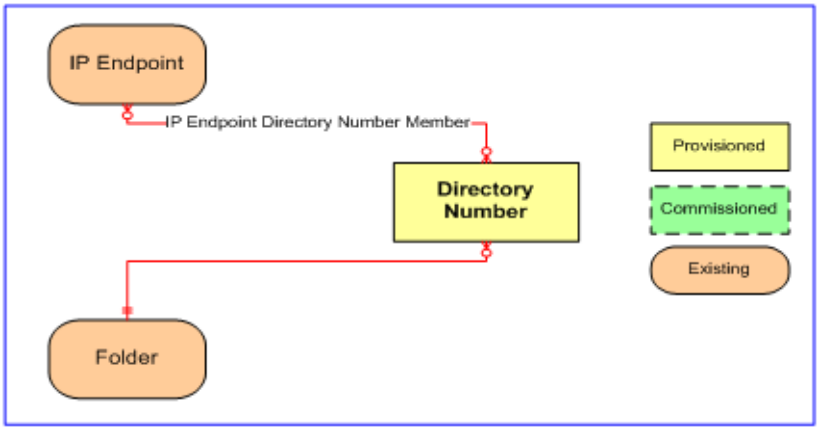
The following fields can be read or set for Directory Number resources:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits From `DimensionItem`				
DirectoryNumberOrPattern	String(50) (n-v pair)	This is a string indicating the dial-able number for the directory number. This can be the actual numeric number or a route pattern with wildcards.	No	Yes
IPCCEnabled	Bool(n-v pair)	This is a flag indicating whether the directory number is IPCC enabled. For example; in a Cisco UCCE deployment model this value is set if there is a matching device target on the Unified CCE associated with the Unified Communications Manager to which this directory number belongs. This matching is done based on the value in the `CONFIG_PARAM` column in `TB_DIM_DIRECTORY_NUMBER` and the `DIRECTORY_NUMBER_OR_PATTERN` in for this record. The default value is false (the directory number is not IPCC-enabled).	No	Yes
RoutePartitionBizUrn	Int(n-v pair)	This is the route partition associated with the directory number.	No	No

Associated Name Fields for Directory Numbers

As well as the fields above, the following associated name fields can also be used to sort Directory 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.

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



Directory number relationships

REST Protocol

The following table is a summary for the Directory Number REST API.

- **Base URL:** `https://<server>:8085/resourcemanagement/rest/resources`
- **ID Format**: JSON

Operation	Mode	Method	Status	URL
Audit	Sync.	GET	200 OK	[Base URL]/directory-numbers/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/directory-numbers
Delete	Async.	DELETE	202 Accepted	[Base URL]/directory-numbers/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/directory-numbers
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/directory-numbers/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/directory-numbers/
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_DIRECTORY_NUMBER&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/directory-numbers/<id>,<id>

Operation	Mode	Method	Status	URL
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/directory-numbers				

Enterprise Skillgroup Resource Item

The `EnterpriseSkillgroup` represents a specific collection of skillgroups.

Fields

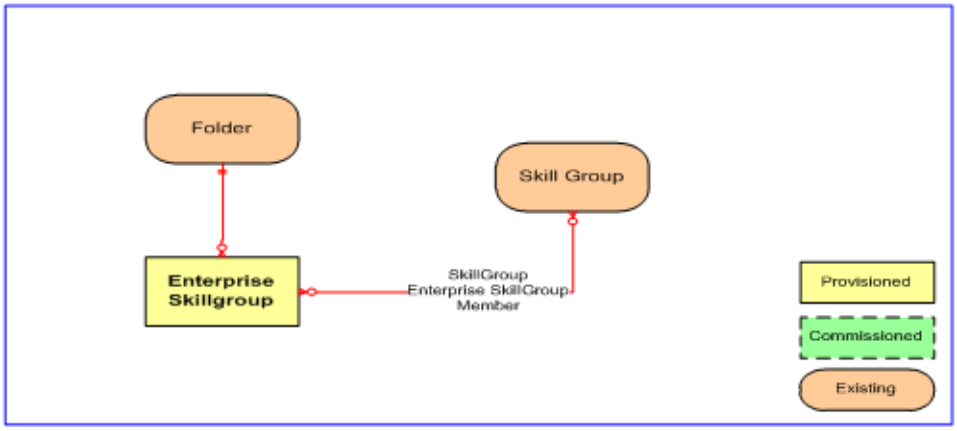
The following fields can be read or set for Enterprise Skillgroup resources:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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

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.

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



Enterprise skillgroup relationships

REST Protocol

The following table is a summary for the Enterprise Skillgroup REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format**: 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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/enterprise-skillgroups				

Expanded Call Variable Resource Item

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

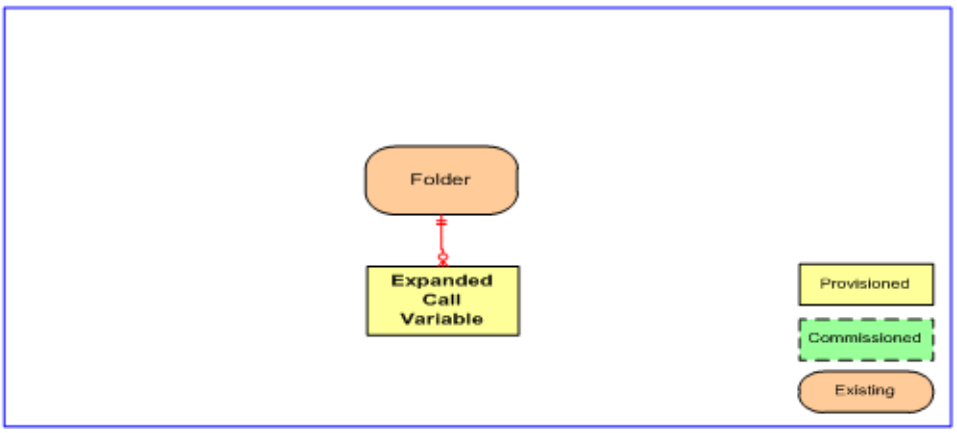
Fields

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

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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

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



Expanded call variable relationships

REST Protocol

The following table is a summary for the Expanded Call Variable REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** JSON

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]/
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/expanded-call-variables				

IP Endpoint Resource Item

The `IPEndpoint` resource represents the termination point of any IP stream. This includes devices such as phones as well as other IP sinks. This API is only used when Unified CCMP is the only provisioning tool; when there are other provisioning tools, Unified CCMP detects this type by reading the switch directly.

Fields

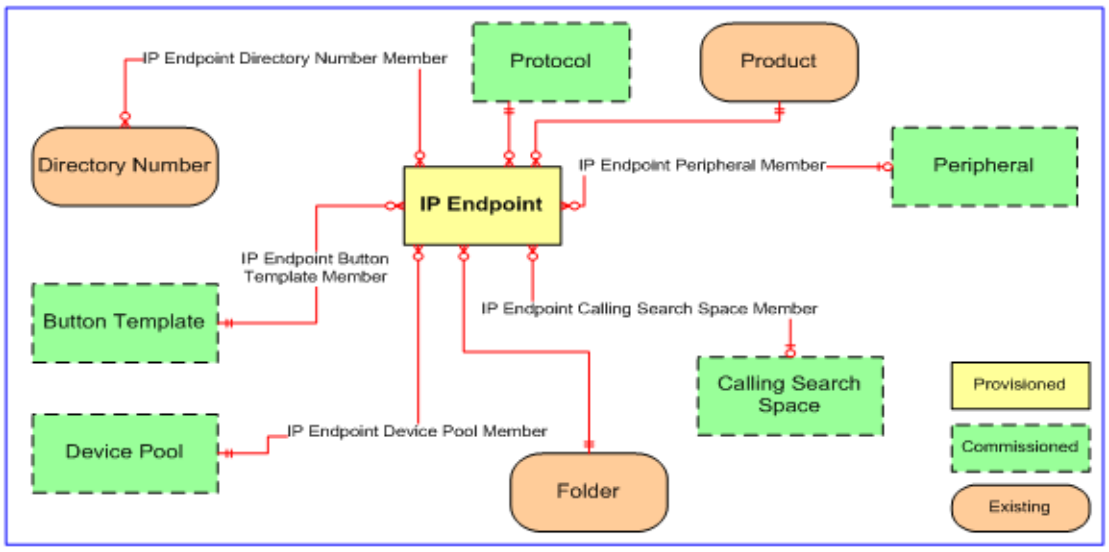
The following fields can be read or set on IP Endpoint Resource objects.

Element Name	Data Type	Description	Required?	Pkey?
Inhabits From `DimensionItem`				
PeripheralUrn	String(10) (n-v pair)	The peripheral or equipment identifier on which the IP Endpoint will be located. Note: cannot be updated after the Create.	No	No
DevicePoolUrn	String(10) (n-v pair)	The device pool in which this IP Endpoint will be located. Note: cannot be modified using the Update call.	Yes	No
CallingSearchSpaceUrn	String(10) (n-v pair)	The Calling Search Space in which this IP Device will be located.	No	No
IPEndPointButtonTemplateUrn	String(10) (n-v pair)	The button template to be associated with this device. Note: this is only applicable to devices which support button templates.	Yes, if the device supports button templates, no otherwise.	No
ProductCodeBizUrn	String(10) (n-v pair)	Foreign key to a product code. Code type: `CG_DEVICE_PRODUCT`. Cannot be updated after creation.	Yes	No
ProtocolCodeBizUrn	String(10) (n-v pair)	Foreign key to a protocol code. Code type: `CG_DEVICE_PROTOCOL`. Cannot be updated after creation.	Yes	No
MACAddress	String(17) (n-v pair)	The MAC address of the device.	Yes	No
TemplateData	Xml(n-v pair)	The template data to be used when creating this IP device. Cannot be updated after creation.	Yes	No
ExtensionMobilityEnabled	Bool(n-v pair)	Flag indicating whether the extension mobility is to be used with this IP Device.	No	No

Associated Name Fields for IP Endpoints

As well as the fields above, the following associated name fields can also be used to sort IP Endpoint 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.

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



IP endpoint relationships

REST Protocol

The following table is a summary for the IP Endpoint REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** JSON

Operation	Mode	Method	Status	URL
Audit	Sync.	GET	200 OK	[Base URL]/ip-endpoints/<id>,<id>/audits?fromDate=<fromdate>&toDate=<toDate>…
Create	Async.	POST	202 Accepted	[Base URL]/ip-endpoints
Delete	Async.	DELETE	202 Accepted	[Base URL]/ip-endpoints/<id>,<id>

Operation	Mode	Method	Status	URL
Describe	Sync.	GET	200 OK	[Base URL]/meta/ip-endpoint
Move	Sync.	PUT	200 OK	[Base URL]/<destinationid>
Retrieve	Sync.	GET	200 OK	[Base URL]/ip-endpoints/<id>,<id>
Save	Sync.	POST	200 OK	[Base URL]/members/ip-endpoints
Search	Sync.	GET	200 OK	[Base URL]?queryString=type%3aIT_IP_ENDPOINT&max%3a10
Update	Async.	PUT	202 Accepted	[Base URL]/ip-endpoints/<id>,<id>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/resources-ip-endpoints				

Product and Protocol Identities

The `ProductCodeBizUrn` and `ProtocolCodeBizUrn` fields must be specified when creating the IP Endpoint in order for Unified CCMP to know the type of device that needs to be created. For example, an IP Communicator.

To identify the correct codes for use with resources of this type the following search queries can be used through the web service search API:

First of all the identity for the product must be retrieved, then the protocol.

Product

The search query looks like this:

```
type:IT_CODE_GROUP text:"CG_DEVICE_PRODUCT"
```

The search query will return an identity of the correct code group, for example, 2825. This identity can be used to retrieve the code for the IP Communicator Product using a search query like this:

```
type:IT_CODE text:"CISCO_IP_COMMUNICATOR" childof:IT_CODE_GROUP,2825
```

This returns the identity of the IP Communicator product which may then be set to the IP Endpoint `ProductCodeBizUrn` property.

Protocol

The search query looks like this:

```
type:IT_CODE_GROUP text:"CG_DEVICE_PROTOCOL"
```

This search query will return an identity of the correct code group, for example, 1765. This identity can be used to retriever the code for SCCP Protocol using a search query like this:

```
type:IT_CODE text:"SCCP" childof:IT_CODE_GROUP,1765
```

This returns the identity of the IP Communicator protocol which may then be set to the IP Endpoint `ProtocolCodeBizUrn` property.

Auto Selection of Device Pool

When creating a phone through the web service and -1 is specified for the `DevicePoolUrn`, Unified CCMP uses the following logic to automatically select a device pool to associate with the new phone:

1. Lookup the call manager time group to find a time group that corresponds to the user's time zone. This lookup is performed first using the standard name for the time zone, otherwise the time zone offset and daylight savings information is used.
2. Lookup the call manager group as follows:

- If a phone already exists that is associated (via the device pool) to a call manager group with spare capacity (observing the 250 phone capacity limit), then select the call manager group of the existing phone.
- Otherwise, select the call manager group with the lowest available capacity but with enough space for a further 20 (configurable) phones.

3. Select the Device Pool associated with the chosen call manager group and time group.

The following errors may be reported:

- Unable to locate a call manager group with spare capacity.
- Unable to find a time group for the user's time zone.

The capacity limit of 250 phones per call manager group is configured by the data stored in `TB_ADM_CAPACITY`.

IVR Script Resource Item

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 that in the Unified CCMP web application, IVR scripts are called "IVR apps."

Fields

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

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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

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.

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

REST Protocol

The following tables are for the IVR Scripts REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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>
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
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/ivr-scripts				

Label Resource Item

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

Fields

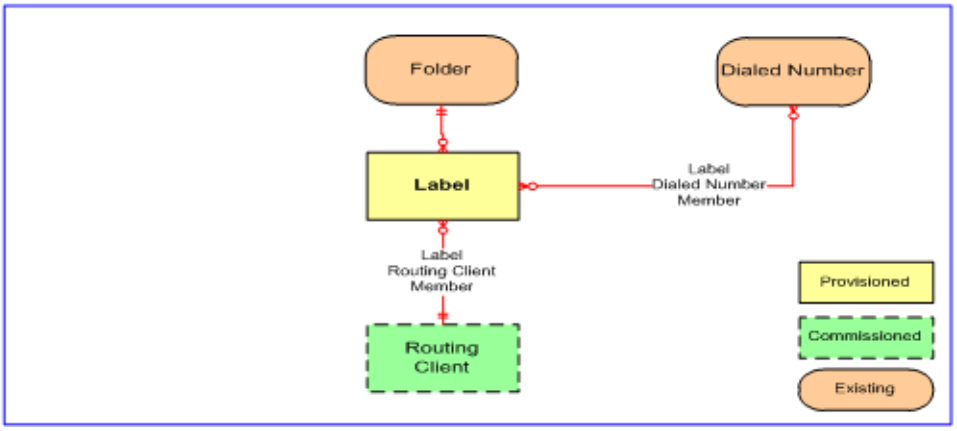
The following fields can be read or set for Label resources:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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	No	Yes

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.

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



Label relationships

REST Protocol

The following table is a summary for the Label REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** JSON

Operation	Mode	Method	Status	URL
Audit	Sync.	GET	200 OK	[Base URL]/labels/<id>,<id>/audits?fromDate=<fromdate>&toDate=<to Date>…
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/labels				

Media File Resource Item

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.

Fields

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

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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

Element Name	Data Type	Description	Required?	Pkey?
Size		The size of the item. This field cannot be created or edited.	N/A	Yes
Yes Checksum		The check-sum of the item. This field cannot be created or edited.	N/A	Yes

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.

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

REST Protocol

The following table is a summary of the Media Files REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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
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

Operation	Mode	Method	Status	URL
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/mediafiles				

Network VRU Script Resource Item

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

Fields

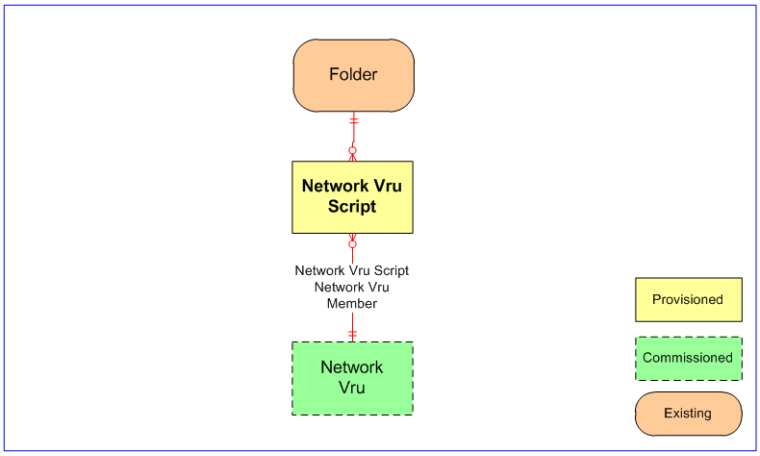
The following fields can be read or set for Network VRU script resources:

Element Name	Data Type	Description	Required?	Pkey?
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
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.	No	Yes
DepartmentBizUrn	Int(n-v pair)	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

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.

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



Network VRU Script Relationships

REST Protocol

The following table is a summary for the Network VRU Script REST API.

- **Base URL**: https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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
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>

Operation	Mode	Method	Status	URL
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-
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/network-vru-scripts				

Person Resource Item

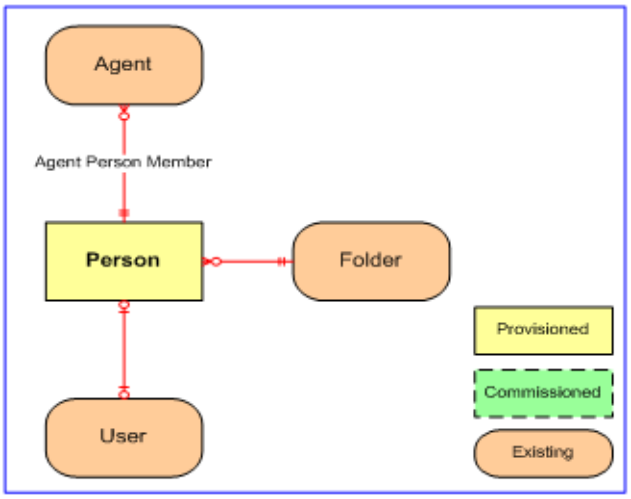
The `Person` resource represents any person resource on the system, not only customer service representatives but also managers and end customers (people not registered on switch) It is often used for a high speed data dip for CRM-style data-directed routing and IVR services.

Element Name	Data Type	Description	Required?	Pkey?
Inhabits From `DimensionItem`				
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
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
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

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.

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



REST Protocol

The following table is a summary for the Person REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format**: 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
Delete	Async.	DELETE	202 Accepted	[Base URL]/persons/<id>,<id>
Describe	Sync.	GET	200 OK	[Base URL]/meta/person

Operation	Mode	Method	Status	URL
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/persons				

Precision Attribute Resource Item Fields

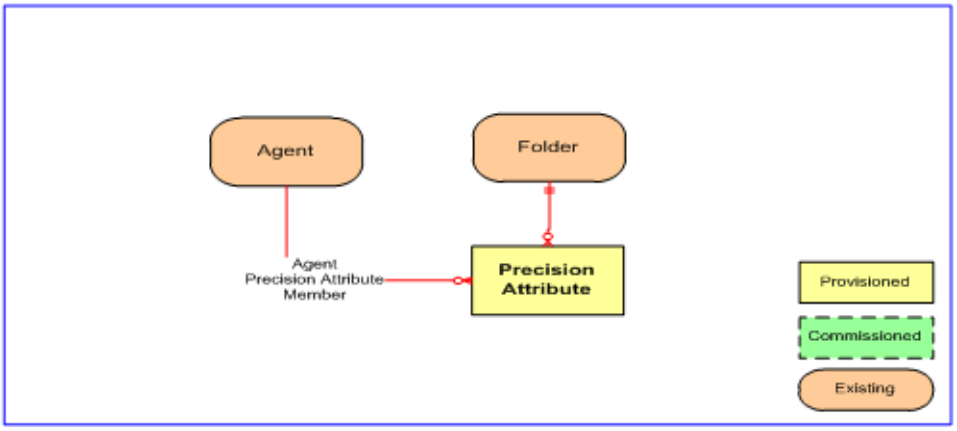
The `PrecisionAttribute` resource represents the attributes that may belong to an Agent. It is used for precision queue based call routing.

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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)	Used only for Proficiency Data Types, and default value is 1.	No	Yes
MaximumValue	String(n-v pair)	Used only for Proficiency data types, and default value is 10.	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

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

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



Precision attribute relationships

Precision Queue Resource Item

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.

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.
PrecisionQueueStepPrecisionAttributeMember	-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.
(PrecisionQueueStepPrecisionAttributeMember)	-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()`.

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

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

2. Select the Precision Queue to be deleted and find its resource key.
3. 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
   ```

4. Create a resource array containing the resource keys of the returned Precision Queue Steps.
5. Add the Precision Queue resource key to the end of the resource array.
6. 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 that you cannot delete the last Precision Queue Step unless you delete the Precision Queue itself.

Precision Queue Resource Item Fields

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

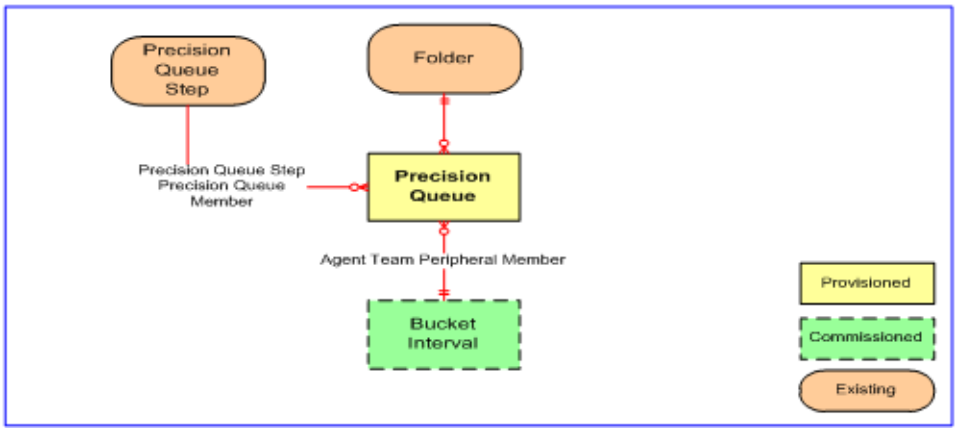
Element Name	Data Type	Description	Required?	Pkey?
Inhabits From `DimensionItem`				
AgentOrdering	Int(n-v pair)	Determines the order of agents in a precision queue sub-queue using the following values: 1 = LAA (agent availability time), 2 = Most skilled agent, 3 = Least skilled agent	No	Yes
CallOrdering	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. Default value is 0	Yes	Yes

Element Name	Data Type	Description	Required?	Pkey?
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 Default value is 1-ignore abandoned calls	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

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.

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



Precision queues

REST Protocol

The following table is a summary for the Precision Queue Resource REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources

- **ID Format:** JSON

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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/precision-queues				

Route Resource Item

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

Fields

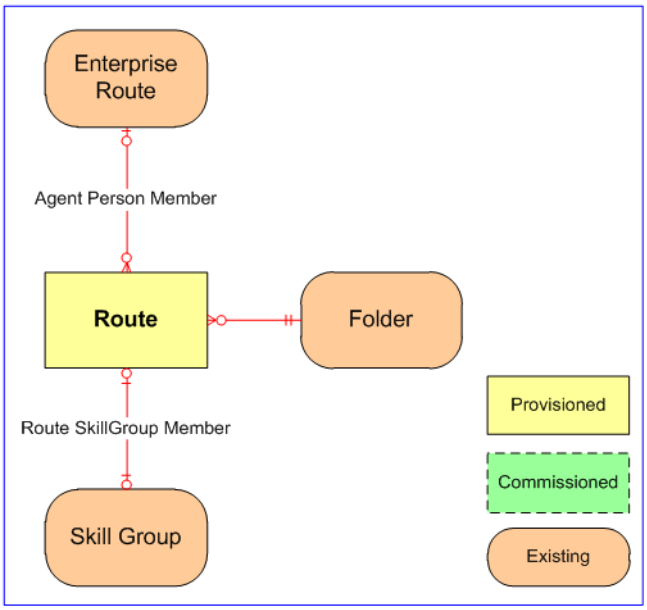
The following fields can be read or set for Route resources.

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

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.

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



REST Protocol

The following table is a summary for the Route Resource REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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

Operation	Mode	Method	Status	URL
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/routes				

Route Partition Resource Item

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

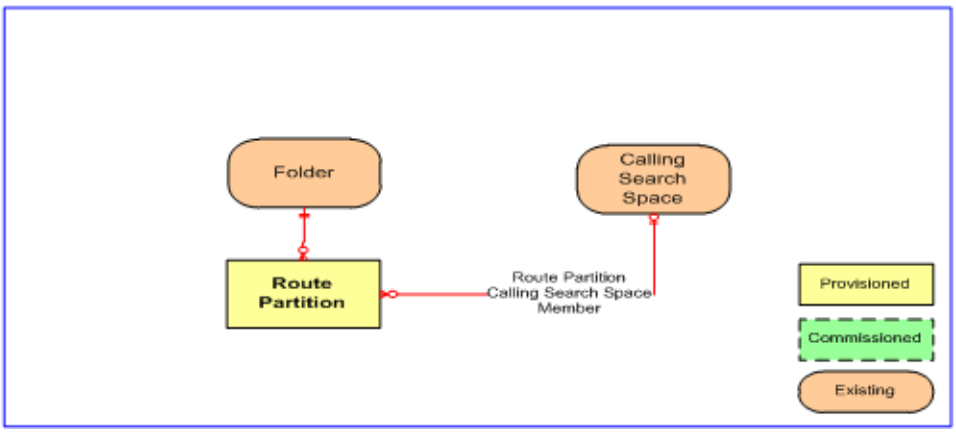
Fields

There are no further fields for this item.

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.

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



Route Partition Relationships

REST Protocol

The following table is a summary for the Route Partition Resource REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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>

Exception	See "Errors" on page 37
Example	POST https://Web01:8085/resourcemanagement/rest/resources/route-partitions

Service Resource Item

The `Service` resource represents a service on a peripheral.

Fields

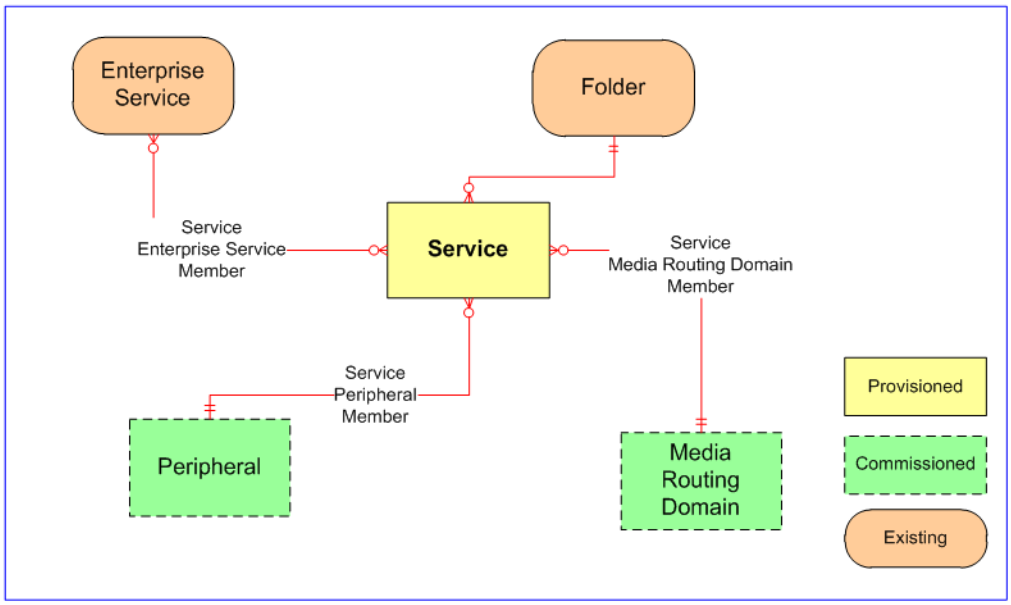
The following fields can be read or set for Service resources:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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 CCMP 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	String(n-v pair)	This is the type of service level used for the service on the peripheral.	No	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.	No	Yes
ServiceLevelThreshold	Int(n-v pair)	This is the threshold in seconds for the service.	No	Yes
Extension	String	This is the extension number for the skill group.	No	Yes
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

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.

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



Service relationships

REST Protocol

The following table is a summary for the Service Resource REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources

- **ID Format**: 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
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>

Exception	See "Errors" on page 37
Example	POST https://Web01:8085/resourcemanagement/rest/resources/services

Skill Group Resource Item

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.

Fields

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

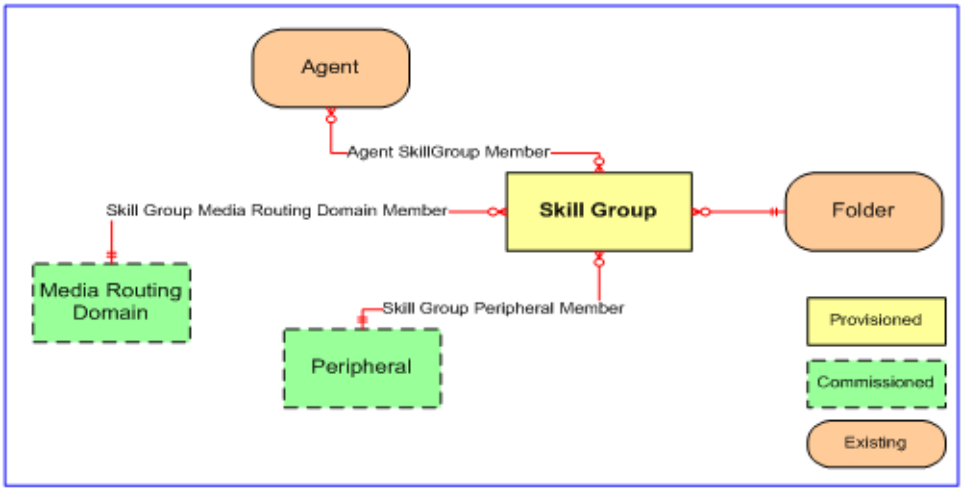
Element Name	Data Type	Description	Required	Pkey
Inhabits From `DimensionItem`				
PeripheralUrn	String(10) (n-v pair)	The peripheral or equipment identifier on which the skill group will be located. If -1 -s supplied then Unified CCMP will pick the Peripheral automatically. Note: cannot be updated after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

Element Name	Data Type	Description	Required	Pkey
PeripheralName	String(50) (n-v pair)	This is the name for the skillgroup as known to the peripheral it is associated with	No	Yes
AvailableHoldoffDelay	Short(n-v pair)	This is the number of seconds before an agent becomes available after a call is terminated. Defaults to 0.	No	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	No	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
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 CCMP 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

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.

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



Skill group relationships

REST Protocol

The following table is a summary for the Skill Group Resource REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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
Delete	Async.	DELETE	202 Accepted	[Base URL]/skillgroups/<id>,<id>

Operation	Mode	Method	Status	URL
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/skillgroups				

Tenant Resource Item

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.

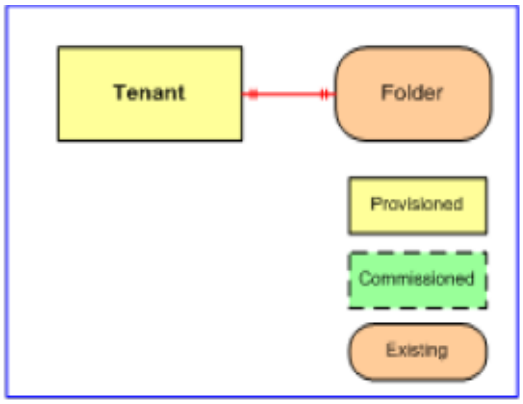
Fields

There are no further fields for this item.

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.

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



Tenant relationships

REST Protocol

The following table is a summary for the Tenant Resource REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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]/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>

Exception	See "Errors" on page 37
Example	POST https://Web01:8085/resourcemanagement/rest/resources/tenants

User Variable Partition Resource Item

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

Fields

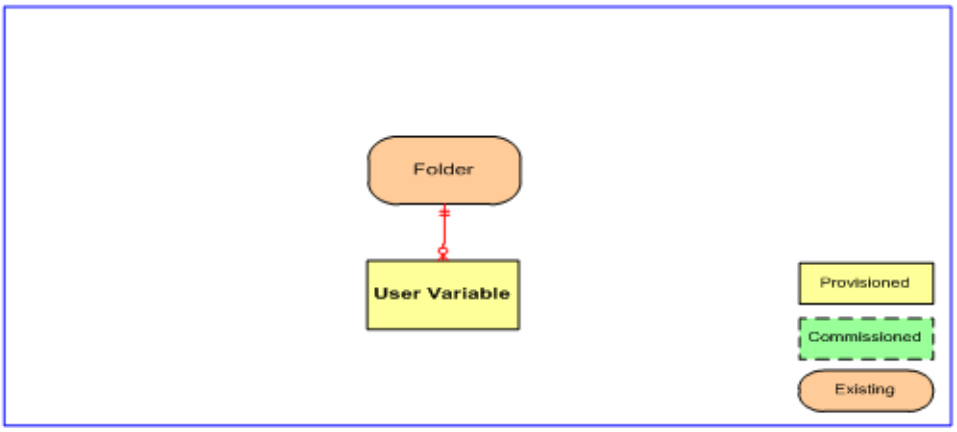
The following fields can be read or set for User Variable resources:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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.	No	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.	No	Yes

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

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



User variable relationships

REST Protocol

The following table is a summary for the User Variable Resource REST API.

- **Base URL:** https://<server>:8085/resourcemanagement/rest/resources
- **ID Format:** 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
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>
Exception See "Errors" on page 37				
Example POST https://Web01:8085/resourcemanagement/rest/resources/user-variables				

Non-Provisionable Remote Resource Types

About Non-Provisionable Remote Resources

These resource types are supported for searching and retrieving only.

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.

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

Resource Management Web Service Member Types

Member types define memberships between entities.

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

Member Type	Internal Name	REST Parameter
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
Code Code Group Member	MT_CODE_CODE_GROUP_MEMBER	code-code-group-member
Device Profile Directory Number Member	MT_DEVICE_PROFILE_DIRECTORY_NUMBER_MEMBER	device-profile-directory-number-member
Device Profile IP Endpoint Member	MT_DEVICE_PROFILE_IP_ENDPOINT_MEMBER	device-profile-ip-endpoint-member
Device Profile Person Member	MT_DEVICE_PROFILE_PERSON_MEMBER	device-profile-person-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
Gadget Provider User Mapping	MT_GADGET_PROVIDER_USER_MAPPING	gadget-provider-user-mapping
Group Group Member	MT_GROUP_GROUP_MEMBER	group-group-member

Member Type	Internal Name	REST Parameter
IP Endpoint Button Template Member	MT_IP_ENDPOINT_BUTTON_TEMPLATE_MEMBER	ip-endpoint-button-template-member
IP Endpoint Calling Search Space Member	MT_IP_ENDPOINT_CALLING_SEARCH_SPACE_MEMBER	ip-endpoint-calling-search-space-member
IP Endpoint Device Pool Member	MT_IP_ENDPOINT_DEVICE_POOL_MEMBER	ip-endpoint-device-pool-member
IP Endpoint Directory Number Member	MT_IP_ENDPOINT_DIRECTORY_NUMBER_MEMBER	ip-endpoint-directory-number-member
IP Endpoint Peripheral Member	MT_IP_ENDPOINT_PERIPHERAL_MEMBER	ip-endpoint-peripheral-member
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

Member Type	Internal Name	REST Parameter
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
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

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.

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
Code Code Group member	True	True	True	True	Code Group	Code
Device Profile Directory Number Member	True	True	True	True	Directory Number	Device Profile
Device Profile IP Endpoint Button Template Member	True	True	True	True	IP Endpoint Button Template	Device Profile
Device Profile Person Member	True	True	True	True	Person	Device Profile
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
Gadget Provider User Mapping	True	True	True	True	User	Gadget Provider
Group Group Member	True	True	True	True	Group	Group
Item Category Member	True	True	True	True	Category	Any Dimension Item

Resource Member Type	Create	Edit	Delete	Read	Parent	Child
IP Endpoint Button Template Member	True	True	True	True	IP Endpoint Button Template	IP Endpoint
IP Endpoint Calling Search Space Member	True	True	True	True	Calling Search Space	IP Endpoint
IP Endpoint Device Pool Member	False	False	False	True	Device Pool	IP Endpoint
IP Endpoint Directory Number Member	False	False	False	True	Directory Number	IP Endpoint
IP Endpoint Peripheral Member	False	False	False	True	Peripheral	IP Endpoint
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

Resource Member Type	Create	Edit	Delete	Read	Parent	Child
Skill Group Service Member	True	True	True	True	Service	Skill Group
User Group Member	True	True	True	True	Group	User

Member Pkey Types

Parent	Child
Agent Team	Agent
Peripheral	Agent
Precision Attribute	Agent
Skill Group	Agent
Routing Script	Call Type
Code Group	Code
Directory Number	Device Profile
Call Type	Dialed Number
Directory Number	IP Endpoint
Precision Attribute	Precision Queue
Campaign	Query Rule
Calling Search Space	Route Partition
Campaign	Skill Group
Service	Skill Group

REST Protocol

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.

- Base URL: https://<server>:8085/resourcemanagement/rest/resources

- ID Format: JSON

Operation	Mode	Method	Status	URL
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>
Exception See "Errors" on page 37				
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				

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.

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.

Associated Name Field	Related Element
ParentCreatedByLoginName	Entry.CreatedById

Agent Agent Desktop Member

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.

Fields

This class has no further fields.

Agent Agent Team Member

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

Fields

The AgentAgentTeamMember class has the following fields:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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 that at least one of `Supervisor`, `PrimarySupervisor`, and `PhysicalMember` must be set.

Agent Desktop Dialed Number Member

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.

Fields

There are no further fields for this item.

Agent Peripheral Member

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

Fields

There are no further fields for this item.

Agent Precision Attribute Member

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

Fields

The AgentPrecisionAttributeMember class has the following fields:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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

Agent Team Dialed Number Member

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.

Fields

This item has no further fields.

Agent Skill Group Member

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

Fields

The AgentSkillgroupMember class has the following fields:

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

Call Type Routing Script Member

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

Fields

The CallTypeRoutingMember class has the following fields:

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

Code Code Group Member

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

Fields

There are no further fields for this item.

Device Profile Directory Number Member

The DeviceProfile Directory Number Member defines a membership between a Device Profile and a Directory Number. It is a many-one relationship and each Device Profile can only be associated one Directory Number.

Fields

The `DeviceProfileDirectoryNumberMember` class has the following fields:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits From `DimensionMember`				
LineIndex	Int(n-v pair)	This is a 0 based index for the Directory Numbers associated with this Device Profile. This must be unique for a particular Device Profile.	No	Yes
MaxNumCalls	Int(n-v pair)	The maximum number of calls to the directory number.	No	Yes
BusyTrigger	Int(n-v pair)	The configuration per line appearance and per cluster for a Directory number after which the call to that specific Directory Number is rejected with a cause of busy.	No	Yes

Device Profile IP Endpoint Button Template Member

The Device Profile IP Endpoint Button Template Member defines a relationship between a Device Profile and an IP Endpoint Button Template. It is a many-one relationship and each Device Profile can only be associated with one IP Endpoint Button Template.

Fields

There are no further fields for this item.

Device Profile Person Member

The Device Profile Person Member defines a relationship between a Device Profile and Person. It is a many-one relationship and each Device Profile can only be associated with one Person.

Fields

There are no further fields for this member class.

Dialed Number Call Type Member

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

Fields

The `DialedNumberCallTypeMember` class has the following fields:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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.	No	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

Dialed Number Routing Client Member

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.

Fields

There are no further fields for this class.

Dialed Number Media Routing Domain Member

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.

Fields

There are no further fields for this class.

Gadget Provider User Mapping

The Gadget Provider User Mapping defines a membership between a Gadget Provider and an Unified CCMP User. It stores provider-specific and user-specific credentials required for the user to access the gadget provider. It is a many-many mapping.

Fields

The GadgetProviderUserMapping class has the following fields:

Element Name	Data Type	Description	Required?
Inhabits From `Entry`			
UserId	String(n-v pair)	GUID of user to which this mapping applies.	Yes
GadgetProviderId	String (n-v pair)	GUID of gadget provider to which this mapping applies.	Yes
Domain	String(n-v pair)	The domain needed by the authorization type.	No
Password	String(n-v pair)	The password of this user on the third party data service	No
AuthType	Char(n-v pair)	The type of authorization supported by the third party data service and which should be used by this deployment. One of:	B: Basic, each API call must have a Basic Authorization Header
S: Session, a login session must be established	W: Windows, when another Windows account must be used for that server, for example, for a data source owned by another provider	C: Custom, a custom authentication scheme is required, for example, a custom authentication header	Yes
Field1	String(n-v pair)	Reserved for future use	No
Field2	String(n-v pair)	Reserved for future use	No

Element Name	Data Type	Description	Required?
Field3	String(n-v pair)	Reserved for future use	No
Enabled	Bool(n-v pair)	Determines if the mapping is enabled for use. If disabled then the credentials will not be used for Gadget Provider authentication.	Yes

Group Group Member

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.

Fields

There are no further fields for this class.

Item Category Member

The Item Category Member defines a membership between a Category and a dimension item (for example, Agent, Call Type, Skill Group). It is a many-many relationship.

Fields

There are no further fields for this class.

IP Endpoint Button Template Member

The IP Endpoint Button Template Member defines a membership between an IP Endpoint and a Button Template. It is a many-one relationship and each IP Endpoint can only be associated with one Button Template.

Fields

There are no further fields for this class.

IP Endpoint Device Pool Member

The IP Endpoint Device Pool Member defines a relationship between an IP Endpoint and a Device Pool. It is a many-one relationship and each IP Endpoint can only be associated with one Device Pool.

Fields

There are no further fields for this class.

IP Endpoint Directory Number Member

The IP Endpoint Directory Number Member defines a relationship between an IP Endpoing and a Directory Number. It is a many-many relationship.

Fields

There are no further fields for this class.

IP Endpoint Directory Number Member

The IP Endpoint Directory Number Member defines a relationship between an IP Endpoint and a Directory Number. It is a many-many relationship.

Fields

There are no further fields for this class.

IP Endpoint Directory Number Member

The IP Endpoint Directory Number Member defines a relationship between an IP Endpoint and a Directory Number. It is a many-many relationship.

Fields

The `IPEndpointDirectoryNumberMember` class has the following fields:

Element Name	Data Type	Description	Required?	Pkey?
Inhabits From `DimensionMember`				
LineIndex	Int(n-v pair)	This is a 0 based index for the Directory Number associated with the IP Endpoint (Phone). This must be unique for a particular IP Endpoint.	No	Yes
MaxNumCalls	Int(n-v pair)	The maximum number of calls to the directory number. The default is 2.	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
BusyTrigger	Int(n-v pair)	The configuration per line appearance and per cluster for a Directory number after which the call to that specific Directory Number is rejected with a cause of busy. The default is 1.	No	Yes
TemplateData	Xml(n-v pair)	The template data.	No	Yes

IP Endpoint Peripheral Member

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

Fields

There are no further fields for this class.

Label Dialed Number Member

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

Fields

There are no further fields for this class.

Label Routing Client Member

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.

Fields

There are no further fields for this class.

Network Vru Script Network Vru Member

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.

Fields

There are no further fields for this class.

Precision Queue Bucket Interval Member

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.

Fields

There are no further fields for this class.

Precision Queue Step Precision Attribute Member

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.

Fields

The `PrecisionQueueStepPrecisionAttributeMember` class has the following fields.

Element Name	Data Type	Description	Required?	Pkey?
Inhabits 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.	No	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.	No	Yes

Element Name	Data Type	Description	Required?	Pkey?
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	No	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

Precision Queue Step Precision Queue Member

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.

Fields

There are no further fields for this class.

Route Skill Group Member

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

Fields

There are no further fields for this class.

Service Enterprise Service Member

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

Fields

There are no further fields for this class.

Service Media Routing Domain Member

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.

Fields

There are no further fields for this class.

Service Peripheral Member

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.

Fields

There are no further fields for this class.

Skill Group Enterprise Skill Group Member

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.

Fields

There are no further fields for this class.

Skill Group Media Routing Domain Member

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.

Fields

There are no further fields for this class.

Skill Group Peripheral Member

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.

Fields

There are no further fields for this class.

Skill Group Service Member

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.

Fields

The SkillgroupServiceMember class has the following fields.

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

User Group Member

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

Fields

There are no further fields for this class.

Resource Management Web Service APIs

Create

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.

Parameters

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

Element Name	Data Type	Description	Required?
resources	Resource[]	An array of resources to be provisioned to the Contact Center environment. Note: The Identity field should be set to -1 for each new resource being created.	Yes

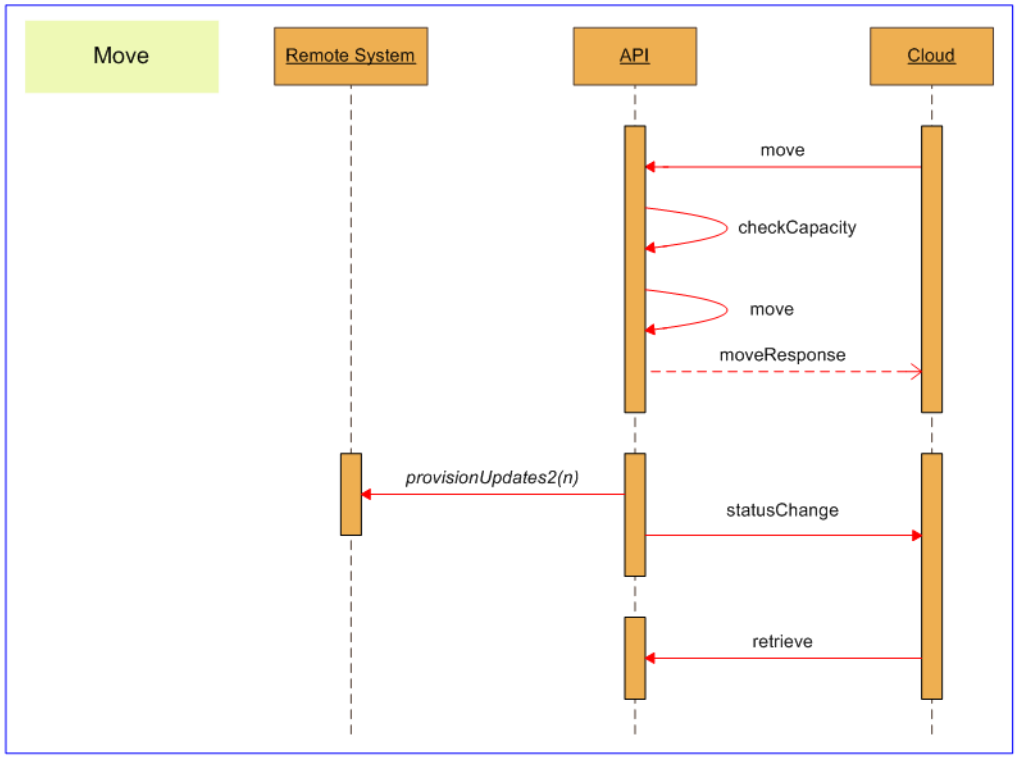
Return Type

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

Element Name	Data Type	Description	Required?
resources	Resource[]	An array of resources to be provisioned to the Contact Center environment. Note: The Identity field should be set to -1 for each new resource being created.	Yes

REST Protocol

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



Sequence Diagram for Resource Management Create API

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.

Update

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

Parameters

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

Element Name	Data Type	Description	Required?
Resources	Resource[]	An array of resources to be provisioned to the Contact Center environment.	Yes

Return Type

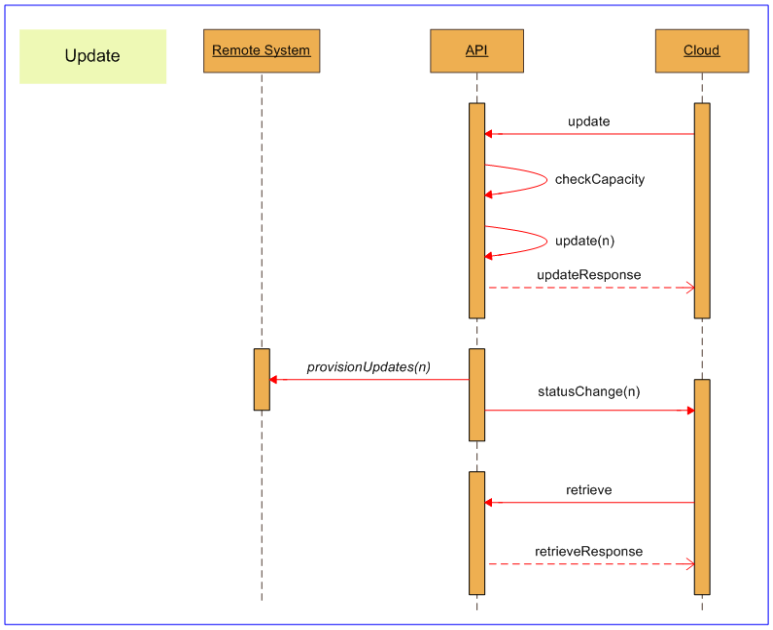
The `Update()` API returns the following objects.

Data Type	Description	Required?
RequestResult[]	An array of result from the resources to be updated.	Yes

REST Protocol

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

Data Type	Description
Input/ Output format	JSON
Example	https://APPSRV01:8085/ResourceManagement/rest/Resources/agents/315553



Sequence diagram for Resource Management Update API

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"Creating Items with Pkey Maps" on page 148.

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

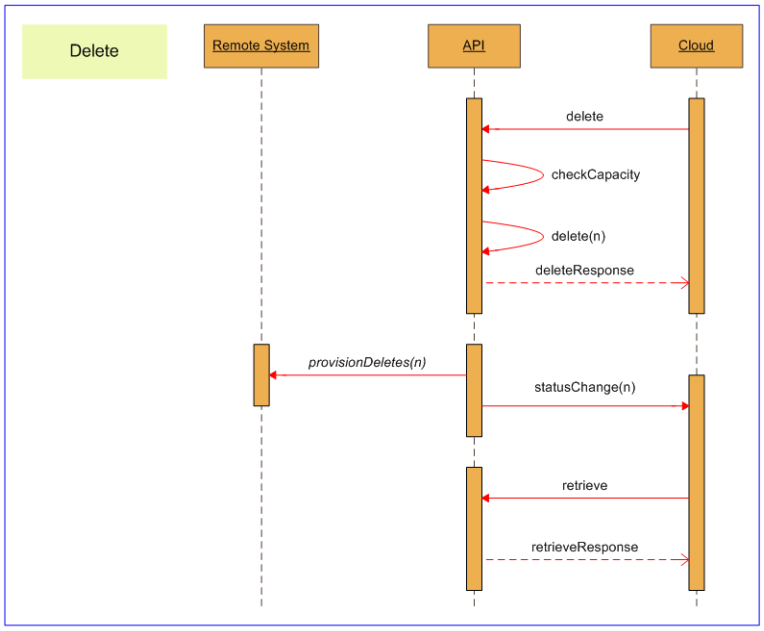
Return Type

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

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

REST Protocol

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	JSON
Example	https://APPSRV01:8085/ResourceManagement/rest/Resources/agents/315553



Sequence diagram for Resource Management Delete API

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 that the last pkey map for a resource or membership cannot be deleted - you must delete the item itself.

Save

The `Save()` API enables you to create, update, and delete multiple items and types in a single API call. Use the `Save()` API when:

- You want 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).

Parameters

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

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 that 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

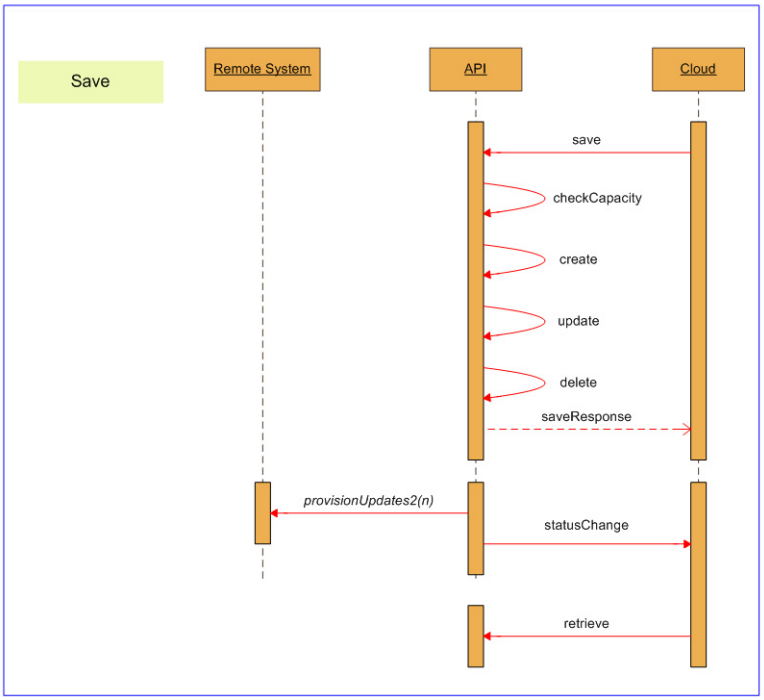
Return Type

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

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

REST Protocol

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	JSON
Example	https://APPSRV01:8085/ResourceManagement/rest/Resources/members/folders



Sequence diagram for Resource Management Save API

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 the underlying Create, Update, or Delete option.

Move

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.

Parameters

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

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 "Resource Key" on page 50.	Yes

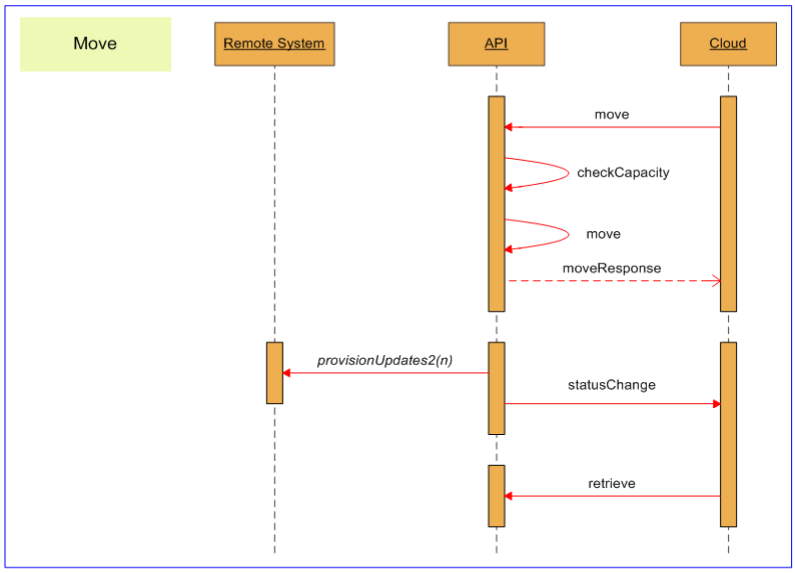
Return Type

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

Data Type	Description	Required?
RequestResult[]	An array of results from the resources to be moved.	Yes

REST Protocol

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



Sequence diagram for Resource Management Move API

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.

Retrieve

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.

Parameters

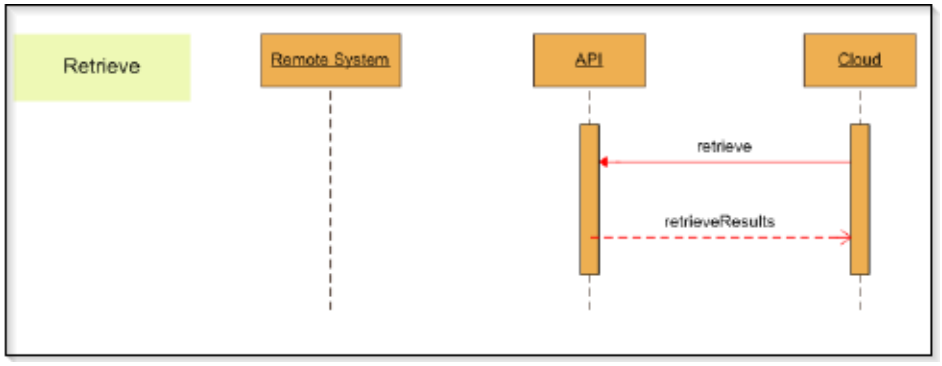
The Retrieve API expects the following parameters:

Element Name	Data Type	Description	Required?
resourceKeys	ResourceKey[]	An array of resource identities for the items to be retrieved. For more information, see "Resource Key" on page 50	Yes

Return Type

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

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. For more information, see "Common Fields and Classes" on page 43.	Yes



Sequence diagram for Resource Management Retrieve API

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 also contains the equipment-specific details for the earliest equipment mapping that is still active.
- 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.

Search

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.

Parameters

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

Element Name	Data Type	Description	Required?
queryString	String	The search string containing the search terms that specify the items to be retrieved.	Yes
excludeFilter	ResourceKey[]	A collection of ResourceKey objects for items to be excluded from the search results. For more information, see "Common Fields and Classes" on page 43.	No

Search String Syntax

The search string is constructed from one or more of search terms. Table 6.159 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 then apply to the entire search string. Global search terms are marked with an asterisk “*” 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 A
childof:"Agent Team",1234gent Team Member"
```

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.

Search Terms

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 that 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 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.

Search Term	Description	Syntax and Examples
		`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.
effectiverange	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.
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.
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.

Search Term	Description	Syntax and Examples
memberchildof	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.
memberparentof	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.
modified	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.
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.
		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. Include one of the other search terms to further restrict the result set, 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.
		`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.

Search Term	Description	Syntax and Examples
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.
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.

Search Term	Description	Syntax and Examples
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.
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”.

Search Term	Description	Syntax and Examples

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:`

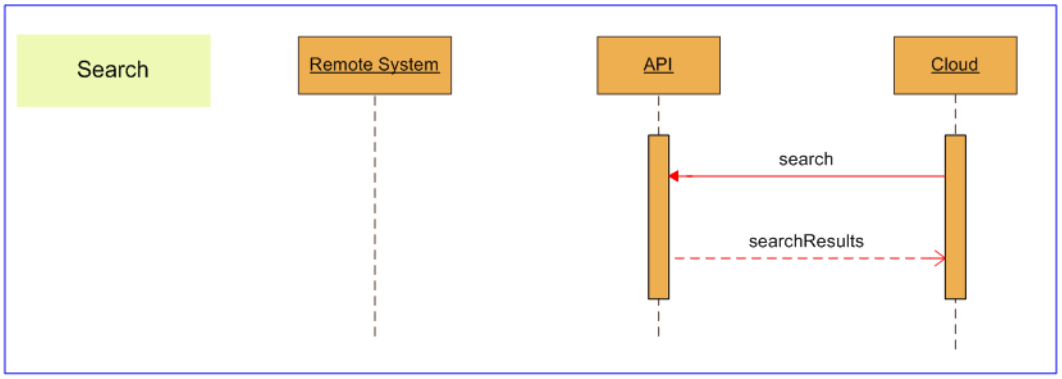
Return Type

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

Data Type	Description	Required?
Resource[]	The results of the search as a collection of Resource objects.	Yes

REST Protocol

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



Sequence diagram for Resource Management Search API

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 information, see "Retrieving Items with Pkey Maps" on page 155.

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.

Parameters

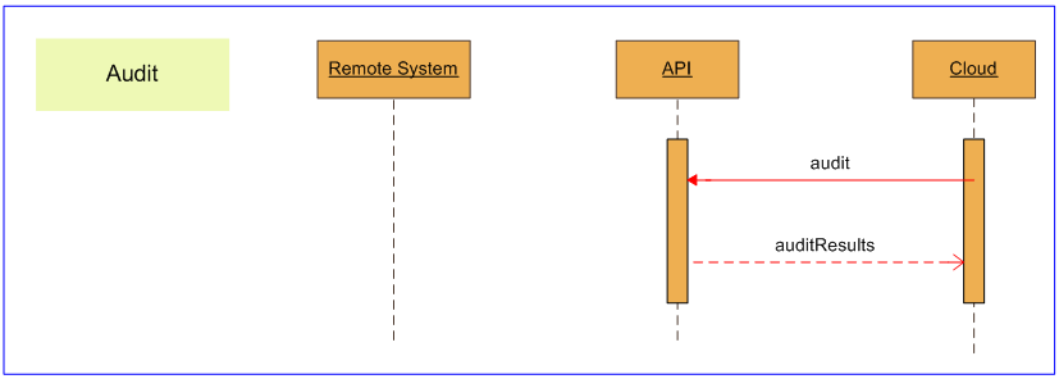
The Audit() API expects the following parameters:

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

Element Name	Data Type	Description	Required?
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 field in each of the returned audit records.	Yes

REST Protocol

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	JSON
Example	https://APPSRV01:8085/ResourceManagement/rest/resources/agents/315553/audits?fromDate=2011-10-04T00%3a00%3a00Z&toDate=2012-06-06T10%3a43%3a57Z



Sequence diagram for Resource Management Audit API

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.

Upload

The `Upload()` API is a generic API that enables 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.

Parameters

The Upload() API expects the following parameters:

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.	Yes
AdditionalData	AdditionalData	The additional data required for the upload. Depends on the resource type. 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:

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

Additional Data		
Name	**Value**	**Required?**
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

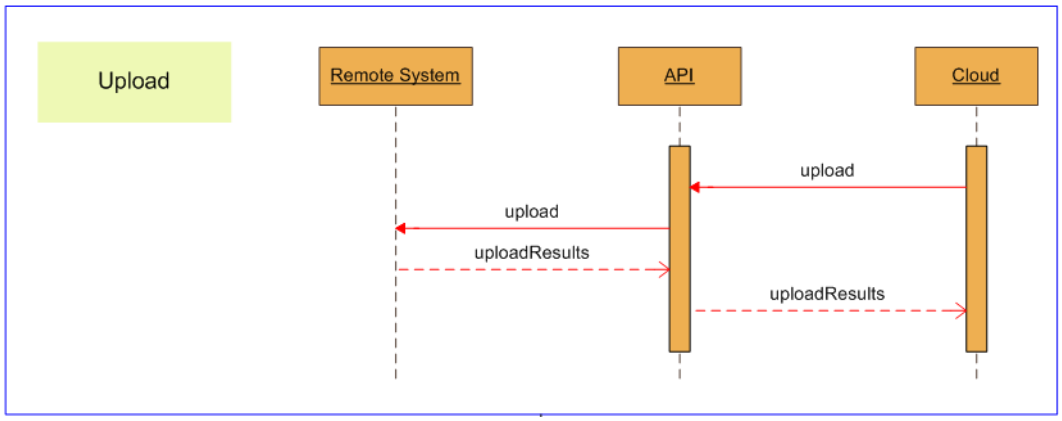
Return Type

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

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

REST Protocol

Data Type	Description
URL	https://<server>:8085/ResourceManagement/rest/resources/<resource type>s/<id>/content?contentType=<contentType>
HTTP Method	PUT
Input/ Output format	JSON
Example	https://<server>:8085/ResourceManagement/rest/resources/mediafiles/3565/content?contentType=audio/wav
Header	Content-Type: multipart/form-data; boundary="----------8d08f053d6a90ad" Content-Length: 216829
Part 1: Additional DataFormat: JSON	----------8d08f053d6a90ad Content-Disposition: form-data; name="additionaldata" Content-Type: application/json; charset=utf-8{"Fields":[{"Name":"Servers","Value":"21,25,29"}]}
Part 2: ContentFormat: 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).



Sequence diagram for Resource Management Upload API

Download

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.

Parameters

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

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.	Yes

The supported resource types and the associated content types are:

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

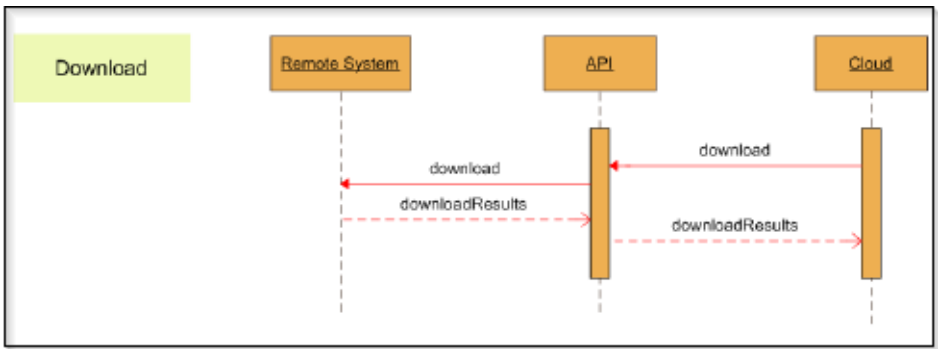
Return Type

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

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

REST Protocol

Data Type	Description
URL	https://<server>:8085/ResourceManagement/rest/resources/<resource type>s/<id>/content?contentType=<contentType>
HTTP Method	GET
Input/ Output format	JSON
Example	https://<server>:8085/ResourceManagement/rest/resources/mediafiles/4893/content?contentType=audio/wav



Sequence diagram for Resource Management Download API

Deploy

The `Deploy()` API is a generic API that enables 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 CCMP, 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

Parameters

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

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.	Yes
AdditionalData	AdditionalData	Additional data required for the deployment. Depends on the resource type. 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:

Resource Type	Content Type	Content Format
IVR Script	application/zip	ZIP file containing one or more VXML applications created by Cisco Call Studio

Additional Data		
Name	**Value**	**Required?**
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

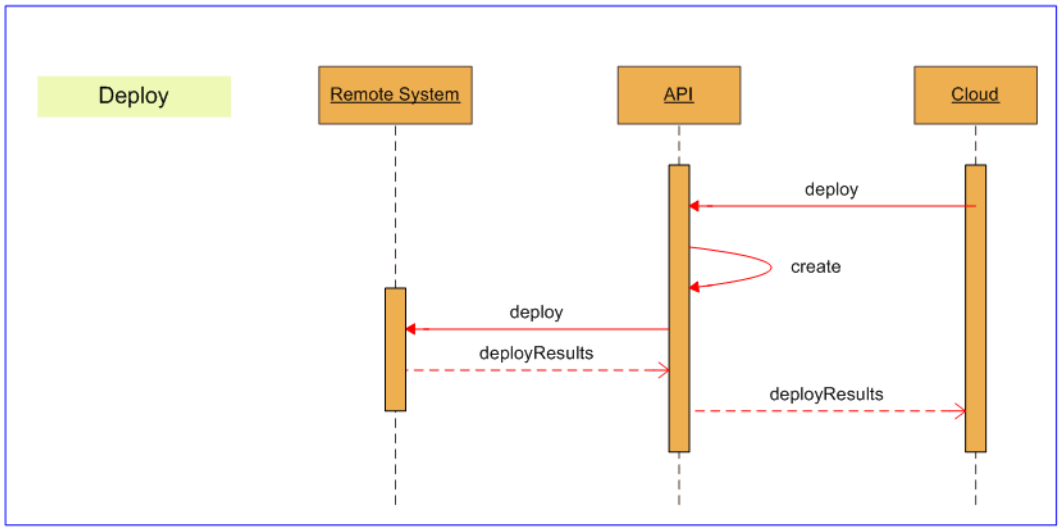
Return Type

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

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

REST Protocol

Data Type	Description
URL	https://<server>:8085/ResourceManagement/resources/<resource type>s/deploy?contentType=<contentType>
HTTP Method	POST
Input/ Output format	JSON
Example	https://<server>:8085/ResourceManagement/rest/resources/ivr-scripts/deploy?contentType=application/zip
Header	Content-Type: multipart/form-data; boundary="----------8d0a38f0cd807bb" Content-Length: 24100
Part 1: Additional DataFormat: 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: ContentFormat: 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).



Sequence diagram for Resource Management Deploy API

Resource Management Web Service Examples

Create Agent

Use Case Sequence

The following prerequisites must be met before creating an Agent:

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

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

Action	API Calls and Parameters
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

REST Protocol

Data Type	Description
CURL	`curl -X POST "https://<server>/Portal/ws/ResourceManagement/rest/resources" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"\",\"BusinessIdentity\":\"\",\"Type\":\"\",\"EffectiveFrom\":\"2020-09-28T16:19:03.149Z\",\"EffectiveTo\":\"2020-09-28T16:19:03.149Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	POST
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources`

Update Agent

Use Case Sequence

The following prerequisites must be completed before updating an Agent.

- The Create Agent use case sequence has been executed without any existing agent desktop, skill groups and team memberships.
- The caller of the Unified CCMP 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.

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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>or retrieve Id = Agent URN

REST Protocol

Data Type	Description
CURL	`curl -X PUT "https://<server>/Portal/ws/ResourceManagement/rest/resources/agent" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"string\",\"BusinessIdentity\":\"string\",\"Type\":\"Agent\",\"EffectiveFrom\":\"2020-09-28T16:24:19.090Z\",\"EffectiveTo\":\"2020-09-28T16:24:19.090Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	PUT
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/agent`

Delete Agent

REST Protocol

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

Data Type	Description
CURL	`curl -X DELETE "https://<server>/Portal/ws/ResourceManagement/rest/resources/agents/17585,17586" -H "accept: application/json"`
HTTP Method	Delete
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/agents/17585,17586`

Retrieve Agent

REST Protocol

Data Type	Description
CURL	`curl -X POST "https://<server>/Portal/ws/ResourceManagement/rest/resources" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"Resources\": [{ \"Identity\": \"string\", \"BusinessIdentity\": \"string\", \"Type\": \"Agent\", \"EffectiveFrom\": \"2020-09-18T16:01:16.058Z\", \"EffectiveTo\": \"2020-09-18T16:01:16.058Z\", \"Status\": \"S\", \"Changestamp\": 6, \"Fields\": {}, \"CustomFields\": {} }]}"`
HTTP Method	POST
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/%28agent%29`

Search For Agent

REST Protocol

Data Type	Description
CURL	`curl -X GET "https://<server>/Portal/ws/ResourceManagement/rest/resources?queryString=agent" -H "accept: application/json"`
HTTP Method	GET
Input/Output Format	JSON
Examples	`https://<server>/Portal/ws/ResourceManagement/rest/resources?queryString=agent`

Audit Agent and Person

REST Protocol

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

Data Type	Description
CURL	`curl -X GET "https://<server>/Portal/ws/ResourceManagement/rest/resources/Agent/audits" -H "accept: application/json"`
HTTP Method	GET
Input/Output Format	JSON
Examples	`https://<server>/Portal/ws/ResourceManagement/rest/resources/Agent/audits`

Create Agent Team

Use Case Sequence

The following prerequisites must be met prior to creating an Agent Team.

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

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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>orretrieve Id = Agent Team

REST Protocol

The following request creates an Agent Team called “test team”.

Data Type	Description
CURL	`curl -X POST "https://<server>/Portal/ws/ResourceManagement/rest/resources" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"Test team\",\"BusinessIdentity\":\"Team\",\"Type\":\"AgentTeam\",\"EffectiveFrom\":\"2020-09-28T16:19:03.149Z\",\"EffectiveTo\":\"2020-09-28T16:19:03.149Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	POST
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources`

Update Agent Team

Use Case Sequence

The following prerequisites must be met before updating an Agent Team.

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

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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>orretrieve Id = Agent Team URN

Add/Remove Agent Team Members

The following prerequisites must be met before adding or removing Agent Team members:

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

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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType> or retrieve Id = Agent Team URN

Action	API Calls and Parameters
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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType> or retrieve Id = Agent Team URN

Create Call Type

Use Case Sequence

The following prequisites must be met before creating a call type:

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

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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType> or retrieve Id = Call Type URN

REST Protocol

Data Type	Description
CURL	`curl -X POST "https://<server>/Portal/ws/ResourceManagement/rest/resources/call-types" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"NewCalls\",\"BusinessIdentity\":\"Calltype\",\"Type\":\"Call\",\"EffectiveFrom\":\"2020-09-28T19:02:22.801Z\",\"EffectiveTo\":\"2020-09-28T19:02:22.801Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	POST
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/call-types`

Update Call Type

Use Case Sequence

The following prerequisites must be met before updating a call type:

- 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.

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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType> or retrieve Id = Call Type URN

Add/Remove Routing Script Members

Use Case Sequence

The following prerequisites must be met before adding or removing a routing script member:

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

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
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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>or retrieve Id = Call Type URN

Create Dialed Number

Use Case Sequence

The following prerequisites must be met before creating a dialed number.

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCMP 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.

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

REST Protocol

Data Type	Description
CURL	`curl -X POST "https://<server>/Portal/ws/ResourceManagement/rest/resources/dialed-numbers" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"testDialedNumber\",\"BusinessIdentity\":\"Numbers\",\"Type\":\"Dialed-Numbers\",\"EffectiveFrom\":\"2020-09-28T19:02:22.801Z\",\"EffectiveTo\":\"2020-09-28T19:02:22.801Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	POST
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/dialed-numbers`

Update Dialed Number

Use Case Sequence

The following prerequisites must be met before updating a dialed number:

- The Create Dialed Number use case sequence has been executed without any existing agent memberships.
- The caller of the Unified CCMP 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.

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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>orretrieve Id = Dialed Number URN

Create Directory Number

Use Case Sequence

The following prerequisites must be met before creating a directory number:

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

Actions for Create Directory Number Use Case

Action	API Calls and Parameters
Select the folder in which the directory 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
Create the directory number located in the selected folder. The request will be validated and then queued returning its new directory number URN.	create Directory Number
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>orretrieve Id = Directory Number URN

REST Protocol

Data Type	Description
CURL	`curl -X POST "https://<server>/Portal/ws/ResourceManagement/rest/resources/dialed-numbers" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"-1\",\"BusinessIdentity\":\"Numbers\",\"Type\":\"Dialed-Numbers\",\"EffectiveFrom\":\"2020-09-28T19:02:22.801Z\",\"EffectiveTo\":\"2020-09-28T19:02:22.801Z\",\"Status\":\"R\",\"Changestamp\":0,\"Fields\":\"testDirectoryNumber\",\"CustomFields\":{}}]}"`
HTTP Method	POST
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/dialed-numbers`

Update Directory Number

Use Case Sequence

The following prerequisites must be met before updating a directory number:

- The Create Directory Number use case sequence has been executed without any existing IP Endpoint memberships.
- The Directory Number and associated resources or members are all in the Ready state.

Actions for Update Directory Number Use Case

Action	API Calls and Parameters
Retrieve the Directory Number record	retrieve Id = Call Type URN
Search for the IP Endpoint to which this Directory Number will be associated..	search type:"IP Endpoint" latest:1 status:R
Update the call type. The Directory Number resource status will go from Ready to Synchronizing until provisioned.	create IP Endpoint Directory Number Member
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>orretrieve Id = Directory Number URN

Create Folder

REST Protocol

Data Type	Description
CURL	`curl -X POST "https://<server>/Portal/ws/ResourceManagement/rest/resources/folder" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"-1\",\"BusinessIdentity\":\"Numbers\",\"Type\":\"Dialed-Numbers\",\"EffectiveFrom\":\"2020-09-28T19:02:22.801Z\",\"EffectiveTo\":\"2020-09-28T19:02:22.801Z\",\"Status\":\"R\",\"Changestamp\":0,\"Fields\":\"testDirectoryNumber\",\"CustomFields\":{}}]}"`
HTTP Method	POST
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/folder`

Update Folder

The Update API may be used for the movement of items between different folders within the Unified CCMP 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 CCMP identity for the folder to which the resource should be moved.

REST Protocol

Data Type	Description
CURL	`curl -X PUT "https://<server>/Portal/ws/ResourceManagement/rest/resources/folders/1562" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"string\",\"BusinessIdentity\":\"string\",\"Type\":\"Folder\",\"EffectiveFrom\":\"2020-09-28T19:34:56.244Z\",\"EffectiveTo\":\"2020-09-28T19:34:56.244Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	PUT

Data Type	Description
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/folders/1562`

Create Group

Use Case Sequence

The following prerequisites must be met before creating a group:

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

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

REST Protocol

Data Type	Description
CURL	`curl -X POST "https://<server>/Portal/ws/ResourceManagement/rest/resources/groups/1562" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"string\",\"BusinessIdentity\":\"string\",\"Type\":\"Group\",\"EffectiveFrom\":\"2020-09-28T19:34:56.244Z\",\"EffectiveTo\":\"2020-09-28T19:34:56.244Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	POST

Data Type	Description
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/groups/1562`

Update Group

Use Case Sequence

The following prerequisites must be met before a group can be updated:

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

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

REST Protocol

Data Type	Description
CURL	`curl -X PUT "https://<server>/Portal/ws/ResourceManagement/rest/resources/groups/1562" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"string\",\"BusinessIdentity\":\"string\",\"Type\":\"Group\",\"EffectiveFrom\":\"2020-09-28T19:34:56.244Z\",\"EffectiveTo\":\"2020-09-28T19:34:56.244Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	PUT
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/groups/1562`

Create IP Endpoint

Use Case Sequence

The following prerequisites must be met before creating an IP endpoint:

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

REST Protocol

Data Type	Description
CURL	`curl -X POST "https://<server>/Portal/ws/ResourceManagement/rest/resources/ip-endpoints" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"string\",\"BusinessIdentity\":\"string\",\"Type\":\"IP Endpoint\",\"EffectiveFrom\":\"2020-09-28T19:02:22.801Z\",\"EffectiveTo\":\"2020-09-28T19:02:22.801Z\",\"Status\":\"R\",\"Changestamp\":0,\"Fields\":\"testDirectoryNumber\",\"CustomFields\":{}}]}"`
HTTP Method	POST
Input/Output Format	JSON
Request URL Example	`https://<server>/Portal/ws/ResourceManagement/rest/resources/ip-endpoints`

Update IP Endpoint

Use Case Sequence

The following prerequisites must be met before an IP endpoint can be updated:

- The Create IP Endpoint use case sequence has been executed without any existing agent memberships.
- The caller of the Unified CCMP Web Services is logged in as the tenant administrator and only has access to the agents that the tenant administrator account allows.

The IP Endpoint and associated resources or members are all in the Ready state.

Actions for Update IP Endpoint Use Case

Action	API Calls and Parameters
Retrieve the IP Endpoint record.	retrieve Id = Skillgroup URN
Modify the field(s) of interest.	
Update the IP Endpoint. The IP Endpoint resource status will go from Ready to Synchronizing until provisioned.	update IPEndpoint

REST Protocol

Data Type	Description
CURL	`curl -X PUT "https://prometheus2019.exony.local/Portal/ws/ResourceManagement/rest/resources/ip-endpoints/1562" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"string\",\"BusinessIdentity\":\"string\",\"Type\":\"IP Endpoint\",\"EffectiveFrom\":\"2020-09-28T19:34:56.244Z\",\"EffectiveTo\":\"2020-09-28T19:34:56.244Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	PUT
Input/Output Format	JSON
Request URL Example	`https://prometheus2019.exony.local/Portal/ws/ResourceManagement/rest/resources/ip-endpoints/1562`

Create Person

Use Case Sequence

The following prerequisites must be met before a Person can be created:

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

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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType> or retrieve Id = Person

REST Protocol

Data Type	Description
CURL	`curl -X POST "https://prometheus2019.exony.local/Portal/ws/ResourceManagement/rest/resources/persons" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"string\",\"BusinessIdentity\":\"string\",\"Type\":\"IP Endpoint\",\"EffectiveFrom\":\"2020-09-28T19:02:22.801Z\",\"EffectiveTo\":\"2020-09-28T19:02:22.801Z\",\"Status\":\"R\",\"Changestamp\":0,\"Fields\":\"testDirectoryNumber\",\"CustomFields\":{}}]}"`
HTTP Method	POST
Input/Output Format	JSON
Request URL Example	`https://prometheus2019.exony.local/Portal/ws/ResourceManagement/rest/resources/persons`

JSON Example

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

```
{ "
Resources": [{
"Identity": "-1",
"Type": "Person",
"EffectiveFromRFC3339": "2019-02-28T08:21:57.853Z",
"EffectiveToRFC3339": "2050-02-17T08:21:57.853Z",
"Status": "R",
"Changestamp": 0,
```

```
"RefURL": "",
"Fields": [{
"Name": "FolderId",
"Value": "F9A52F80-4BDE-4992-9A71-93D2490F614A"
},
{
"Name": "Name",
"Value": "John_Smith"
},
{ "
Name": "
LoginName",
"Value": "john.smith"
},
{ "
Name": "
FirstName",
"Value": "John"
},
{ "
Name": "
LastName",
"Value": "Smith"
},
{ "
Name": "
PassPhrase",
"Value": "Pa55word"
} ],
}]
}
```

Update Person

Use Case Sequence

The following prerequisites must be met before a Person can be updated.

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

REST Protocol

Data Type	Description
CURL	`curl -X PUT "https://prometheus2019.exony.local/Portal/ws/ResourceManagement/rest/resources/persons/John" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"Resources\":[{\"Identity\":\"string\",\"BusinessIdentity\":\"string\",\"Type\":\"IP Endpoint\",\"EffectiveFrom\":\"2020-09-28T19:34:56.244Z\",\"EffectiveTo\":\"2020-09-28T19:34:56.244Z\",\"Status\":\"S\",\"Changestamp\":0,\"Fields\":{},\"CustomFields\":{}}]}"`
HTTP Method	PUT
Input/Output Format	JSON
Request URL Example	`https://prometheus2019.exony.local/Portal/ws/ResourceManagement/rest/resources/persons/John`

Create Skill Group

Use Case Sequence

The following prerequisites must be met before a Skill Group can be created:

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

Update Skill Group

Use Case Sequence

The following prerequisites must be met before updating a Skill Group:

- The Create Skill Group use case sequence has been executed without any existing agent memberships.
- The caller of the Unified CCMP 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.

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
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType> or retrieve Id = Skillgroup URN

Create Tenant

Use Case Sequence

The following prerequisites must be met before a Tenant can be created:

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

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
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. 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

Update Tenant

User Case Sequence

The following prerequisites must be met before a Tenant can be updated:

- The Create Tenant use case sequence has been executed.
- The caller of the Unified CCMP 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.

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.	

Action	API Calls and Parameters
Update the Tenant. The Tenant resource status will go from Ready to Synchronizing until provisioned.	update Tenant
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>orretrieve Id = Tenant URN

Create User

User Case Sequence

The following prerequisites must be met before a User can be created.

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

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

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`.

Update User

Use Case Sequence

The following prerequisites must be met before a User can be updated:

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

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

Agent to Skill Group Membership

JSON Example

The following request creates a membership between an agent and a skill group with the addition of RefURL and MembershipType.

```
{
"Resources":
[
{
"Identity": "-1",
"Type": "Agent SkillGroup Member",
"Status": "R", "RefURL": "",
"Fields" :[
{"Name":"ParentId","Value":"33"}
,{"Name":"ChildId","Value":"41"}
,{"Name":"MembershipType","Value":"1"}
]
}
]
}
```

Dialed Number to Call Type Membership

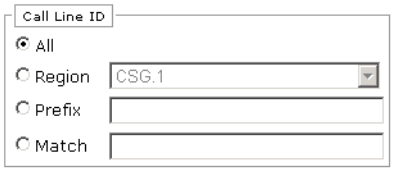
The Dialed Number Call Type membership has a number of fields available on it and may be configured in different ways for Dialed Numbers that belong to Voice or non-Voice Media Routing Domains.

The fields that may be set when creating or updating a Dialed Number Call Type Membership are:

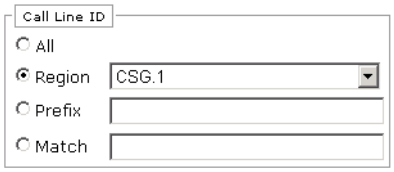
- ParentId (required) the identity of the Call Type resource
- ChildId (required) the identity of the Dialed Number resource
- RegionUrn
- Item (required)
- CLIWildCard
- CLIWildCardType (required)
- CEDWildCardDescription

When creating or editing a Dialed Number connected to a Voice Media Routing Domain, the configuration can be specified as follows.

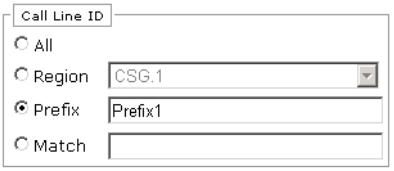
Call Line ID Configuration




```
CliWildCardType 5
CliWildCard Null
RegionUrn Null
```

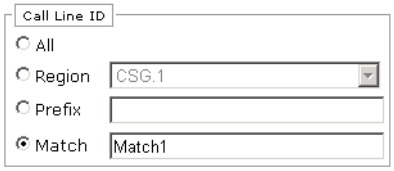



```
CliWildCardType 4
CliWildCard Null
RegionUrn The Identity of the selected Region
```



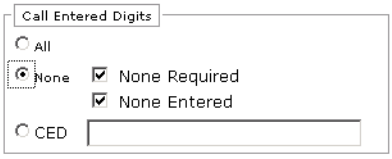

```
CliWildCardType 6
```

```
CliWildCard Prefix1
RegionUrn Null
```

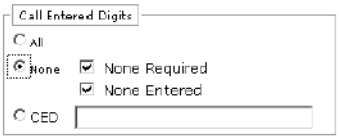



```
CliWildCardType 3
CliWildCard Match1
RegionUrn Null
```

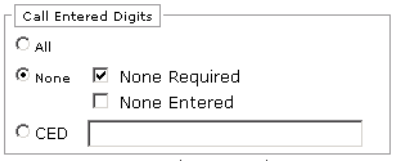
Call Entered Digits Configuration



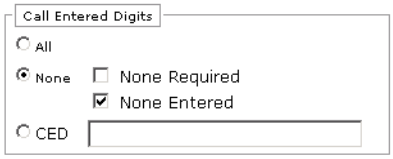

```
CedWildCard _A
```



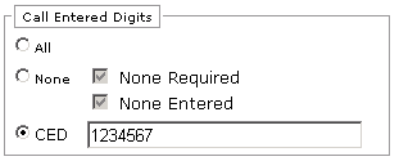

```
CedWildCard _N
```




```
CedWildCard _NR
```




```
CedWildCard _NE
```




```
CedWildCard 1234567
```

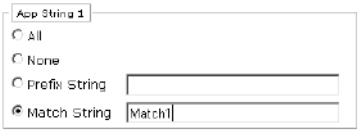
App String 1 Configuration

When creating or editing a Dialed Number connected to a non-Voice Media Routing Domain the configuration can be specified as follows.


```
CedWildCard: _NE
```


```
CliWildCardType 3
CliWildCard Null
RegionUrn Null
```


```
CliWildCardType 6
CliWildCard PrefixString
RegionUrn Null
```

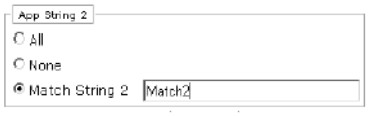



```
CliWildCardType 3
CliWildCard Match1
RegionUrn Null
```

App String 2 Configuration


```
CedWildCard _A
```


```
CedWildCard Null
```




```
CedWildCard Match2
```

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 - when ready, the **Save** button may be pressed to save the changes to the Unified CCMP database and the remote system.

The web service API interaction for this example is as follows:

1. Retrieve the list of Skill Groups that the user can see. The Call to Search API passing in the search string looks like this:

   ```
   type:SkillGroup folder:/Tenant latest:1 max:50 status:R
   ```

2. The user then 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
   ```

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
   ```

4. The Peripheral identity (4321) is used to get the list of possible agents that may be added to the Skill Group.

Call to the Search API with the search string:

```
childof:Peripheral,4321 type:Agent
```

The API is also passed a collection of ResourceKeys for the selected Agents in order to exclude them from the list.

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 CCMP database.
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 CCMP 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.

User Creation

The following example describes the usage of the Web Service APIs to create a new Unified CCMP 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 CCMP (for example, when a new Unified CCE Customer Definition is imported from Unified CCE), three 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:

1. Retrieve the Folder ID 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
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.
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.

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`.
5. The new user has now been created and added to the Advanced Users group.

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:

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.

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
   ```

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.
4. The Dialed Numbers have now been moved from the unallocated location to the tenant folder.

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:

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
```

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
   ```

3. Add the relevant dialed number(s) to the relevant call type(s) as shown in section "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.

4. The new dialed number call type memberships will be added to the Unified CCMP database and their membership URNs returned to the caller for tracking purposes.

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, use 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, use either:

   ```
   parentof:"Dialed Number",<dialed number urn> type:"Call Type" latest:1
   ```

   or

   ```
   memberbychild:"Dialed Number",<dialed number urn>, "Dialed Number Call Type
   Member"
   ```

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:

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
   ```

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
```

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..
4. The existing dialed number call type memberships are marked for deletion in the Unified CCMP database and are provisioned in due course. Note that the memberships are purged from the underlying Unified CCE in this case but will remain in the Unified CCMP database with a status of Deleted for reporting and tracking purposes.
5. Subscribers who have subscribed to this resource type receive a notification for each step in the state machine cycle until the operation is either successful or fails provisioning.

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:

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 returns 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.

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.

Create Agent with Pkey Maps, Method 1 (Explicit Pkey Maps)

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.

Note that you can use separate pkey maps if an item has different details on different remote equipment instances. Additionally, this method may be mixed 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.

Use Case Sequence

The following prerequisites must be met before using this method:

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

Actions for Create Agent with Pkey Maps, Method 1 (Explicit Pkey Maps) Use Case

Action	API Calls and Parameters
Create the Agent, specifying the Agent details and pkey maps.	create Agent
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>or retrieve Id = Agent Id

Create Agent Team with Pkey Maps, Method 2 (Implicit Pkey Maps)

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.

Use this method if an item has the same details of 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.

Use Case Sequence

The following prerequisites must be met before using this method:

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

Actions for Create Agent Team, Two Pkey Maps, Method 2 (Combined Pkey Maps) Use Case

Action	API Calls and Parameters
Create the Agent Team, specifying the Agent Team details and pkey maps.	create AgentTeam
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>or retrieve Id = AgentTeam Id

Update Agent (Additional Pkey Map)

This example adds an additional remote equipment mapping to an existing Agent.

User Case Sequence

The following prerequisite must be met before using this method:

- The specified Agent already exists.

Actions for Update Agent (Additional Remote Equipment Mapping) Use Case

Action	API Calls and Parameters
Retrieve the agent record	retrieve Id = Agent Id
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.	update Agent
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>or retrieve Id = Agent Id

Update Agent (Delete a Pkey Map)

This example deletes one of the remote equipment mappings for an Agent.

Note that you cannot delete the last remaining remote equipment mapping unless you delete the Agent.

Use Case Sequence

The following prerequisite must be met before this method can be used:

- The specified Agent already exists and has at least two pkey map items.

Actions for 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.	delete Agent Pkey
If required, Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType>or retrieve Id = Agent Pkey Id

Create Agent to Agent Team Membership (Two Remote Equipment Mappings)

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.

Use Case Sequence

The following prerequisites must be met before this method can be used:

- 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.

Actions for 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.	update Agent

Create Precision Queue

This example creates a Precision Queue which contains the following:

- 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.

Use Case Sequence

The following prerequisite must be met before this method can be used:

- The specified Precision Attribute exists.

Actions for 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.	create PrecisionQueue
Begin a poll sequence using search or retrieve.	search item:<Id> type:<ResourceType> or retrieve Id = PrecisionQueueURN resourceType:<ResourceType>

4 Subscriptions Web Service

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 the statues of the item.

The Subscriptions Web Service does not support REST protocol.

Securing Notifications with SSL

About Securing the Subscriptions Web Service

SSL can be used to secure and authenticate communications from the Unified CCMP web server (the server generating the notifications) to the subscriber server (the server running the subscriber client application).

This procedure is optional and is only necessary if you plan on using Unified CCMP, and want to secure and authenticate the communications between the Unified CCMP server and the subscriber server. If you do not plan on using Unified CCMP, or do not need to secure and authenticate these communications, then this procedure is not required.

Important: **In order to use the Unified CCMP Web Services, you must first secure the Web Services themselves as outlined in the Unified CCMP Install Guide. This procedure is required and the Unified CCMP Web Services are not useable until this process has been completed.**

When Unified CCMP is first installed, communications between the Unified CCMP server and the subscriber server are secure with a self-signed certificate called *localhost*. While this certificate is suitable for a single server, it cannot be used to secure a multi-server installation.

To secure and authenticate the communications between the Unified CCMP server and the subscriber server:

1. Obtain and install a suitable digital certificate on the Unified CCMP server.
2. Obtain and install a suitable digital certificate on the subscriber server.
3. Configure the Unified CCMP server endpoint behavior to use the two certificates for communication between the servers.
4. Install the public key of the subscriber server on the Unified CCMP server.
5. Ensure the certificate authority root certificate for the Unified CCMP server certificate is available on the subscriber server.
6. Restart the services on both servers.

Obtaining and Installing a Digital Certificate for the CCMP Server

Important: **When CCMP was installed, the CCMP Web Services should have been secured with SSL (see the CCMP Install Guide). If this has not been completed already, complete that procedure before continuing with the instructions here.**

As part of the process of securing the Unified CCMP web services with SSL, a digital certificate was obtained and installed on the Unified CCMP App/Web server. It is recommended that you use the same digital certificate here, as using the same certificate allows you to skip this step.

If you want to use a different certificate, see the Unified CCMP Install Guide for instructions on obtaining and installing another certificate on the Unified CCMP app/web server, as well as granting access to the network service user.

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 CCMP 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 CCMP 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.

Configure the CCMP Server Endpoint Behavior

To configure the endpoint behavior of the CCMP app/web server to use the server certificates for secure communication between the two servers:

1. Identify the subject distinguished name of the certificate you have installed on the CCMP app/web server. This comes in the form of 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 using the following process:

   a. On the Unified CCMP server, go to the **Start** menu and type **mmc** in the command box to open the Microsoft Management Console (MMC).

   b. Click **File > Add Remove Snap-in**, then click **Certificates**, then the **Add** option.

   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 named of the certificate you have installed on the subscriber server. If the subscriber server is a Windows platform, follow the instructions listed above. Otherwise, consult your platform documentation to find out how to do this.

3. On the Unified CCMP app/web server, navigate to **C://Program Files**Unified CCMP **\application server**. Locate the file **Exony.Reporting.Application.Server.exe.config** and open it in a text editor, such as Notepad. Note that you may want to make a safe copy of this file before making any changes to it.

4. Locate the section that begins with `<behavior name="ClientCertificateBehavior">` and within that section, locate the section that begins with `<clientCredentials>`

5. In the `<clientCredentials>` section, locate the `<clientCertificate>` tag and change the `findValue` attribute to the subject distinguished named of the Unified CCMP 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 `<defaultCertficate>` 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=Newbur, S=berkshire, C=GB"
   ```

   then the contents of the <defaultCertificate> tag may become:

   ```
   <defaultCertificate findvalue="CN=silver, OU=dev, O=exony, L=newbury,
   S=berkshire, C=GB"
   storeLocation="LocalMachine" storenName="My"
   x509FindType="FindBySubjectDistinguishedName"/>
   ```

7. Save the changes to this file.

Installing the Public Key of the Subscriber Server Certificate onto the CCMP Server

Installing the subscriber server certificate's public key onto the Unified CCMP app/web server allows the CCMP app/web server to encrypt communications in the manner required by the subscriber server.

To install the public key of the Subscriber Server Certificate onto the CCMP Server:

1. On the subscriber server, obtain the public key for the subscriber server certificate and save it to file. Consult your platform documentation to find out how to do this.

2. Copy the file containing the public key to the CCMP app/web server.
3. On the Unified CCMP 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.

Installing the Root CA Certificate on the Subscriber Server

When connecting the CCMP app/web server to the subscriber server, the CCMP web services certificate needs to be confirmed as valid by the subscriber server. To confirm it as valid, you must obtain the root certificate of the certificate authority (CA) that issued the CCMP web services certificate and install it on the subscriber server.

Important: **This step is only necessary if the provider of the CCMP 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 CCMP server certificate:

1. Obtain the CA root certificate for the CCMP 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. For more information about this process, consult your platform documentation.

Restarting the Services

To restart the services:

1. Save all configuration files and other changes.
2. On the CCMP app/web server, stop and restart the CCMP System Subscription Web Service.
3. On the subscriber server, stop and restart the application service.

More Information about Subscriptions and Notifications

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 (such as 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.

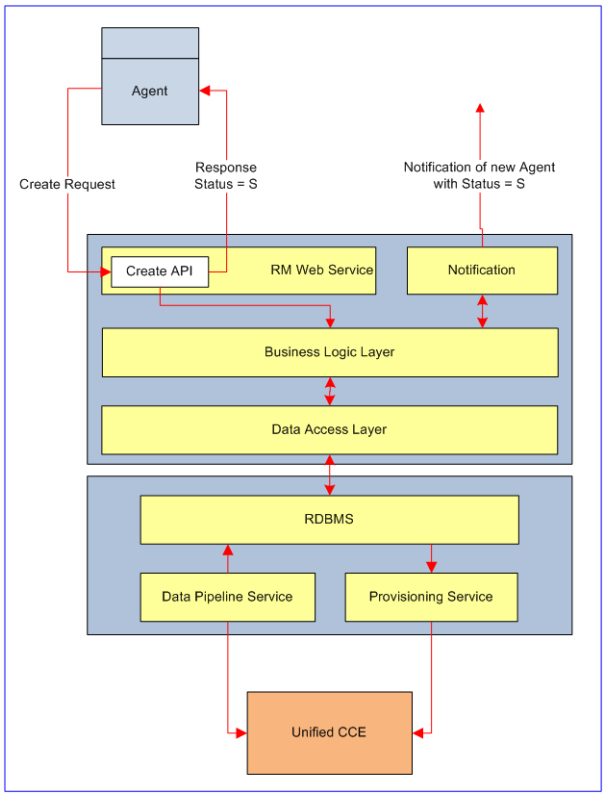
Notification Reporting

Certain resource types require Unified CCMP 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 CCMP 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 CCMP 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). 1. 2. 3.



An example of the provisioning lifecycle.

Not all resource types use the provisioning life-cycle. For some resources, such as User Groups, they are not provisioned on any remote equipment, but are added to Unified CCMP immediately. The notification behavior of these types is subtly different than those that utilize the provisioning lifecycle detailed in the example above.

The following types of notification are supported by Unified CCMP:

- Support Provisioning Life-cycle: has the behavior described above, these are types that rely on Unified CCMP 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 Cycles	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
Directory Number	X		X	X	X
Folder		X	X		X
Group		X	X		X
IP Endpoint	X		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	X		X	X	X
Member			X		X
Agent Skill Group Member	X		X		X
IP Endpoint Directory Number Member	X		X	X	X
User Group Member		X	X		
Group/Group Member		X	X		

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 an Agent 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.

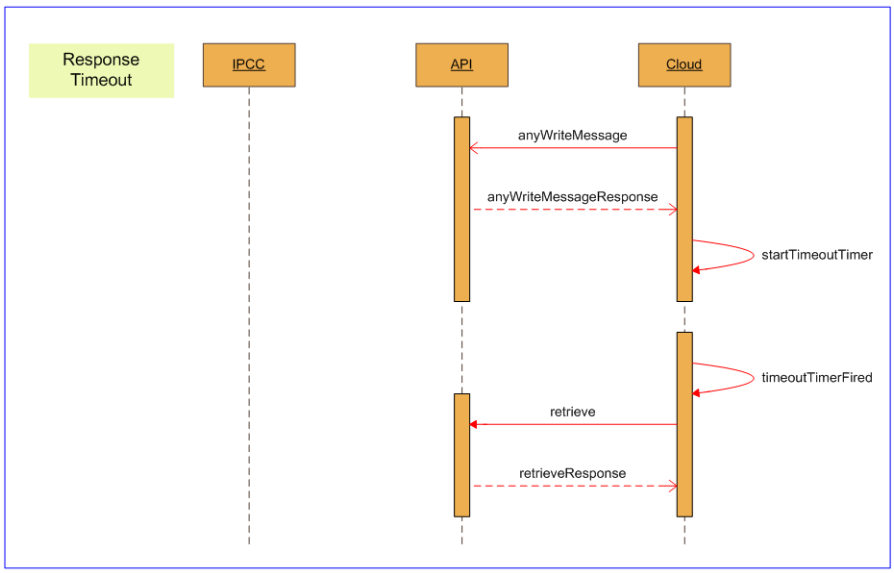
The following subscription method should be applied to protect client applications when notifications are not received, such as when a failure occurs.

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.

API Notification Timeout

If a notification is lost in a situation where receiving a response is critical, the caller must implement a time to retriever the result.



API Notification timeout sequence

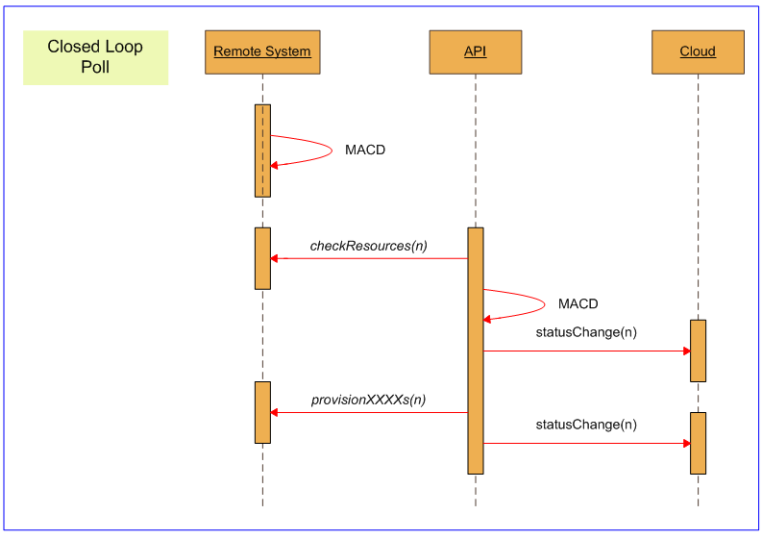
Closed Loop Poll

Moves, adds and changes may be made in the underlying remote equipment. Unified CCMP periodically reads the configuration data from the remote equipment and updates the Unified CCMP 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 CCMP 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.



Closed loop poll sequence diagram

Appendix A: Unified CCMP

Technical Overview

Unified CCMP Management Overview

Unified CCMP 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.

Web Service API Architecture

The Unified CCMP 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/Cisco Unified Communications Manager environment using their exposed software interfaces. For this service, Unified CCMP supports REST web service protocols.

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. Note that the REST protocol cannot be used with this web service.

Web Service API Design Concepts

The Unified CCMP Web Services are hosted within the Unified CCMP 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: The identifier delay is not a problem in practice since the client already has the new identifier. Therefore, it can be used in a subsequent provisioning request, as write requests are always re-loaded from the RDBMS irrespective of the cache state.

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 CCMP 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 CCMP 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.

Appendix B: Valid Time Zones

Valid Time Zones

The following time zones are valid to be specified in the TimeZone field of the User Resource Item. For more information about the User Resource Item and how to use the listed time zones shown here, see "User Resource Item" on page 66.

- 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