



Cisco Packaged Contact Center Enterprise Developers Guide, Release 9.0(x)

First Published: August 02, 2012

Last Modified: March 29, 2013

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2012-2013 Cisco Systems, Inc. All rights reserved.



CONTENTS

PART I

Overview 1

CHAPTER 1

API Usage and Conventions 3

- Change Log 4
- General Usage 4
 - Access 4
 - Object ID 7
 - changeStamp 7
 - Passwords 7
 - HTTP Responses 7
 - API Behavior 9
- Internationalization 9
- Pagination 10
 - Parameters 10
 - Response 10
 - Response Fields 11
 - Important Notes 11
- Permissions Information 12
- Search API 12
 - Overview 12
 - XML Returned 13
 - Default Search Values 14
- Sort API 15
 - Overview 15
 - XML Returned 16
 - Allowable Sort Attributes 16
 - Error Results 19
- Asynchronous API 20

- create 20
- update 20
- delete 21
- Values 21
- Use Cases 21
- Exceptions 24

PART II

The Cisco Packaged Contact Center Enterprise Developers Guide APIs 25

CHAPTER 2

Active Directory Domain API 27

- Active Directory Domain API Commands 27
 - list 27

CHAPTER 3

Agent API 29

- Agent API Commands 29
 - API Parameters 29
 - create 30
 - delete 35
 - list 36
 - get 38
 - update 40
 - Asynchronous API 42

CHAPTER 4

Agent Desk Settings API 43

- Agent Desk Settings API Commands 43
 - API Parameters 43
 - create 44
 - delete 46
 - list 46
 - get 47
 - update 47
 - Asynchronous API 48

CHAPTER 5

Agent State Trace API 49

- Agent State Trace API Commands 49

API Parameters 49

get 49

update 50

CHAPTER 6

Agent Team API 53

Agent Team API Commands 53

API Parameters 53

create 53

delete 55

list 55

get 56

update 57

CHAPTER 7

Attribute API 59

Attribute API Commands 59

API Parameters 59

create 60

delete 61

list 61

get 62

update 62

CHAPTER 8

Bucket Intervals 65

Bucket Intervals API Commands 65

API Parameters 65

create 66

delete 67

list 68

get 68

update 68

Asynchronous API 69

CHAPTER 9

Bulk Job API 71

Bulk Job API Commands 71

API Parameters 71

- create 71
- delete 73
- list 73
- get 73
- Bulk Operations 74
 - Dialed Number Bulk Operation 75
 - Descriptions 75
 - CREATE Operation 75
 - UPDATE Operation 77
 - Agent Bulk Operation 79
 - CREATE Operation 80
 - UPDATE Operation 85

CHAPTER 10

Call Type API 91

- Call Type API Commands 91
 - Parameters for Call Type API 91
 - create 92
 - delete 93
 - list 93
 - get 94
 - update 94
 - Asynchronous API 95

CHAPTER 11

Congestion Control API 97

- Congestion Control API Parameters 97
- Congestion Control get 98
- Congestion Control update 98

CHAPTER 12

Deployment API 101

- Deployment API Commands 101
 - get 101

CHAPTER 13

Deployment Type Info API 103

- Deployment Type Info API Commands 103
 - API Parameters 103

- get 104
- update 105
- VM Validation 108
 - Overview 108
 - VM Checks 108
 - Errors 109

CHAPTER 14**Dialed Number API 111**

- Dialed Number API Commands 111
 - API Parameters 111
 - create 112
 - delete 114
 - list 115
 - get 115
 - update 116
 - Asynchronous API 117

CHAPTER 15**Expanded Call Variable APIs 119**

- Expanded Call Variable API Commands 119
 - API Parameters 119
 - create 120
 - delete 122
 - list 123
 - get 123
 - update 123
 - Asynchronous API 124

CHAPTER 16**Media Routing Domain API 125**

- Media Routing Domain API Commands 125
 - list 125
 - get 126

CHAPTER 17**Network VRU Script APIs 127**

- Network VRU Script API Commands 127
 - API Parameters 127

create 128
 delete 131
 list 131
 get 131
 update 132
 Asynchronous API 132

CHAPTER 18

Precision Queue API 133

Precision Queue API Commands 133
 API Parameters 133
 create 136
 delete 140
 list 141
 get 142
 update 144

CHAPTER 19

Reason Code API 147

Reason Code API Commands 147
 API Parameters 147
 create 147
 delete 148
 list 149
 get 149
 update 149
 Asynchronous API 150

CHAPTER 20

Serviceability API 151

Serviceability API Commands 151
 Introduction 151
 API Parameters 151
 get 152
 System Configuration Validation Rules 154
 System Configuration Validation Output 155
 Serviceability Categories 158
 Capacity 158

CHAPTER 21**Skill Group API 161**Skill Group API Commands **161**API Parameters **161**create **162**delete **164**list **164**get **165**update **166**



PART **I**

Overview

- [API Usage and Conventions, page 3](#)



CHAPTER

1

API Usage and Conventions

The Cisco Packaged Contact Center Enterprise Developers Guide uses [REST](#)-based API functions accessed over HTTP. Five API functions are supported; each is mapped to an HTTP operation. Not all API functions are used for all objects.

The components are:

- create (http POST)—creates an object in the database and returns a response that contains the URL reference to the newly created object. This code sample shows the URL reference returned for a newly-created Bucket Interval: *HTTP/1.1 201 Created Location: <https://192.168.0.1/unifiedconfig/config/bucketinterval/100162>*. The id for the Bucket Interval is *100162*. This URL reference can be used to retrieve the object with an HTTP GET.
- delete (http DELETE)—deletes the object.
- get (http GET)—returns data for an object. For objects for which there are multiple records, GET takes an `<id>`.
- update (http PUT)—modifies an object. For some objects, PUT must include a [changeStamp](#), on [page 7](#), but all other parameters are optional for PUT. *
- list (http GET)—for objects for which there can be multiple records, returns a list.

The POST and PUT operations take a payload for which the input format is XML. GET and DELETE calls do not take a payload. All output is provided as XML when there is a response other than HTTP headers.

XML is case sensitive. When XML data is sent to the server, the tag names must match. `<Name>` and `<name>` are two different XML elements. If a payload contains duplicate fields, only one is transmitted. The duplicates are ignored.

Note: There are two types of URLs used:

- Absolute URLs
- Relative URLs, with `<refURL>` tags

Whereas absolute URLs are used as the target of the POST/PUT/GET/DELETE operations, `<refURL>` tags are sent or returned as part of the XML body.

So, the location header that is returned during a create has the absolute URL format, while the relative URL format is seen in all the XML that is passed in or returned from the REST calls within the `<refURL>` `</refURL>` tag.

The URL format for each type looks like this:

- Absolute URLs: `https://<server_address>/unifiedconfig/config/...`
- Relative URLs, with `<refURL>` tags: `/unifiedconfig/config/...`



Note

The size of the Request Payload for all APIs must be less than or equal to 5 MB.

- [Change Log](#), page 4
- [General Usage](#), page 4
- [Internationalization](#), page 9
- [Pagination](#), page 10
- [Permissions Information](#), page 12
- [Search API](#), page 12
- [Sort API](#), page 15
- [Asynchronous API](#), page 20

Change Log

This section notes the new and changed APIs in this release.

API	See	Notes
Congestion Control	Congestion Control API , on page 97	Re-added into API documentation.
Network VRU Script	Network VRU Script APIs , on page 127	Added routingType field.
Dialed Number	Dialed Number API , on page 111	Cisco Voice MRD is now allowed on dialed numbers with multichannel routing types.

General Usage

Access

Administrators who are in the Active Directory Config Security Group or Setup Security Group have full access to the Cisco Packaged Contact Center Enterprise APIs, unless that access has been limited by the Feature Control Set List Tool and the User List Tool. These tools are Unified CCE Configuration Manager

tools, used together to establish and limit access to the Cisco Packaged Contact Center Enterprise administration tools—both the user interface and APIs—and to Unified CCE Configuration Manager. **Note that the Administrator user name should be in the form of a Fully Qualified Domain Name (FQDN).**

The Feature Control Set List Tool is used to create a Feature Control Set, while the User List Tool associates that Feature Control Set with users. The Feature Control Set List Tool establishes access by marking check boxes for the application names on the Feature Control Set and denies access by leaving the boxes unchecked. The User List Tool can associate a Feature Control Set with a user and/or limit access to read-only.

Full access to a Cisco Packaged Contact Center Enterprise API assumes that permissions are not limited by a Feature Control Set or by a read-only setting on the **User List Tool.**

For example, **if Agent Explorer is unchecked in a Feature Control Set List and the user is associated with that list, the user is restricted from the Agent API.** Likewise, **if Attributes is unchecked in a Feature Control Set List and the user is associated with that list, the user is restricted from the Attribute API.**

Note: A user that is restricted from an API cannot make changes to that API, but the user can still read it.

Note: Most application names on the Feature Control Set List *do not* correspond to Cisco Packaged Contact Center Enterprise APIs. The following table calls out the ones that apply.

API:	Application Name in the Feature Control Set List:
Agent	Agent Explorer
Attribute	Attribute
Agent Desk Settings	Agent Desk Settings List
Precision Queue	Precision Queue
Skill Groups	Skill Group Explorer
Agent Team	List Agent Team
Reason Code	Reason Code List
Bucket Intervals	Bucket Intervals List
Call Type	Call Type List
Dialed Number	Dialed Number/Script Selector List
Expanded Call Variable	Expanded Call Variable List
Network VRU Script	Network VRU Script List
Bulk Job	Dialed Number Bulk Edit AND Agent Bulk Edit
Congestion Control, Deployment Type Info, and Agent State Trace	System Information

It is important to note also that users who have Feature Control Set limitations and read-only cannot see the tools that the Feature Control Set excludes. They can see the other tool, but cannot make changes in those

tools. For example, if Agent Explorer is checked, but read-only is checked for that user in the User List, the user can run Gets and Lists only.

Supervisor Access

In the Webconfig system you can log in either as an Administrator using the Fully Qualified Domain Name (FQDN), or as a Supervisor using the Agent username field.

Supervisors have limited access to and restricted usage of the APIs.

The following table outlines the APIs that Supervisors can access and the associated restrictions. Note that using any methods not listed in the Method column for each API will return a 405 (Method not supported) error. Trying to access any APIs not listed in the table below will return a 404 (Not found) error.

Table 1: Supervisor Access Restrictions

API	Access Level	Method	Additional Restrictions
Agent Team	Read Only	List or get	
Agent	Read and Update	List, get, update	Supervisors that call the Agent list API only see agents on their team(s). Supervisors that try to use the Agent get or update APIs for an agent not on their team(s) get a 404 (Not found) error. When updating an Agent API, Supervisors can change only the following fields: <ul style="list-style-type: none"> • Skill Groups • Default Skill Group • Attributes • Password
Skill Group	Read and Update	List, get, update	Supervisors are only allowed to update the list of agents that are members of the skill group. The supervisor can only add or remove agents from the skill group that are on their team(s).
Attribute	Read Only	List or get	

Precision Queue	Read Only	List or get	
-----------------	-----------	-------------	--

Object ID

Object ID <id>: Using http POST to create all objects generates and returns an *id* for the object.

The DELETE, GET, and PUT operations for these objects are performed using the object id in the REST URL. For example:

- Use this URL to view results for a specified Bucket Interval:
 - `https://<ServerIP>/unifiedconfig/config/bucketinterval/<id>/results`
- Use this URL to delete a Bucket Interval:
 - `https://<ServerIP>/unifiedconfig/config/bucketinterval/<id>`.

Use the List(GET) function to identify the object ids.

```
<results>
<pageInfo>
....
</pageInfo>
<bucketIntervals>
<bucketInterval>...</bucketInterval>
<bucketInterval>...</bucketInterval>
</bucketIntervals>
</results>
```

changeStamp

A changeStamp is a required parameter for the body of a PUT (update) operation for objects.

If you do not provide a changeStamp, the update fails. This mechanism is in place so that two clients cannot edit the record at the same time.

If the update is successful, the database increments the changeStamp by 1.

Passwords

For security, the APIs do not return passwords.

HTTP Responses

All errors are returned as [HTTP 1.1 Status Codes](#). The common codes used by the APIs are:

- **200 OK**: Success
- **201 Created**: The requested item was created.
- **202 Accepted**: The request was accepted. Generally, a URL is provided to obtain additional details, for example, for polling the OAuth status.

- **400 Bad Request:** The request is invalid. Information returned in the ApiErrors message – example below – shows more details.
- **401 Unauthorized:** The authentication credentials were not supplied or were incorrect.
- **403 Forbidden:** Access denied.
- **404 Not Found:** The URI requested does not exist on the server.
- **405 Method Not Allowed:** The method specified in the Request-Line is not allowed for the resource identified by the Request-URI.
- **500 Internal Server Error:** There is a problem on the server. Submit a post to the Forum explaining what you did and the response sent from the server.

Field specific and database errors are provided in an XML error message with the format:

```
<apiErrors>
  <apiError>
    <errorType>Type of Error</errorType>
    <errorData>Field Error Occurred</errorData>
    <errorMessage>A Description of the Error</errorMessage>
    <errorDetail>Extra information about the Error</errorDetail>
  </apiError>
</apiErrors>
```

ErrorDetail Examples

An example of an **errorDetail** field for an error type such as `invalidInput.outOfRange` is as follows:

```
<errorDetail>
  <min>0</min>
  <max>5</max>
</errorDetail>
```

An example of an **errorDetail** field for error types such as `invalidInput.fieldLengthExceeded`, `limitExceeded.expandedCallVariableSize`, and `limitExceeded.totalExpandedCallVariableSize` is:

```
<errorDetail>
  <max>5</max>
</errorDetail>
```



Note

the preceding list of error types is not a comprehensive list and is given only as an example.

ErrorDetail References

Errors of type **referenceViolation** include the following kinds of error detail fields, for example: `totalCount`, `totalShown`, `referenceType`, `name`, `refURL`, `id`, and `deleted`.

For example, if you try to delete a Bucket Interval that is referenced by a Call Type, the error details look like this:

```
<errorDetail>
  <totalCount>1000</totalCount>
  <totalShown>5</totalShown>
  <referenceType>callType</referenceType>
  <references>
    <reference>
      <name>callType1</name>
      <refURL>/unifiedconfig/config/calltype/5000</refURL>
    </reference>
    <reference>
      <name>callType2</name>
      <refURL>/unifiedconfig/config/calltype/5001</refURL>
    </reference>
  </references>
</errorDetail>
```

```

    <reference>
      <name>callType3</name>
      <refURL>/unifiedconfig/config/calltype/5002</refURL>
    </reference>
    <reference>
      <name>callType4</name>
      <refURL>/unifiedconfig/config/calltype/5003</refURL>
    </reference>
    <reference>
      <name>callType5</name>
      <refURL>/unifiedconfig/config/calltype/5004</refURL>
    </reference>
  </references>
</errorDetail>

```

ErrorDetail Script References

For items that are referenced by the script editor, the following detail fields are included for each Master_Script entry: name, id, and versions.

API Behavior

For any field except **list** elements, you can specify the same attribute more than once. However, the API takes the last attribute that you specify for that field.

For example, if you create an Agent using the following XML:

```

<agent>
<agentId>00370</agentId>
<description>bling</description>
<person>
  <firstName>fred</firstName>
  <firstName>bill</firstName>
  <lastName>smithx</lastName>
  <userName>fsmithax</userName>
  <password>freddieboy</password>
  <loginEnabled>true</loginEnabled>
  <changeStamp>0</changeStamp>
</person>
</agent>

```

Notice that `<firstName>` is specified twice:

```

  <firstName>fred</firstName>
  <firstName>bill</firstName>

```

The API takes the second `<firstName>` attribute and sets the Agent's first name to:

```
bill
```

Be aware that this type of behavior is common to all the APIs.

Internationalization

In some of the fields in the APIs, if you enter characters that are not supported by the database, such as native characters, for example, an error is returned that states: **The system does not support these characters.** The fields include `<description>` in any of the APIs that have a `<description>` field, and the following fields in Agent API: `<first.Name>` and `<lastName>`.

For information on how to configure your system to support native character sets, see the latest version of *Installing and Configuring Cisco Packaged Contact Center Enterprise*. Got to: http://www.cisco.com/en/US/products/ps12586/prod_technical_reference_list.html

Pagination

The pagination of the API provides information about how many objects are in the database, as well as pointers to the first, last, previous, and next page of items, if available.

This section outlines the pagination parameters, shows a sample response, and describes the fields that are returned in the response. It also provides important notes about pagination.

Parameters

The following table shows the pagination parameters that you can set.

Parameter Name	Explanation	Notes
startIndex	Specifies the index of the element, at which to start.	Zero-based: 0 is the first element. DEFAULT = 0.
resultsPerPage	Specifies the number of elements to retrieve.	MIN=1. DEFAULT=25 MAX=100.



Note

The following is an example of how to use the pagination parameters when listing a specific element type:

- <https://<server>/unifiedconfig/config/bucketinterval?startIndex=0&resultsPerPage=5>

Response

The following shows an example XML response:

Example XML Response:	<pre> <pageInfo> <resultsPerPage>2</resultsPerPage> <startIndex>0</startIndex> <totalResults>10</totalResults> <firstPage> http://<server>/bucketIntervals/?resultsPerPage=2</firstPage> <lastPage> http://<server>/bucketIntervals/?startIndex=8&resultsPerPage=2</lastPage> <prevPage/> <nextPage> http://<server>/bucketIntervals/?startIndex=2&resultsPerPage=2</nextPage> </pageInfo> <bucketIntervals> <bucketInterval/> <bucketInterval/> </bucketIntervals> </pre>
------------------------------	---

Response Fields

The following table shows the fields that are returned in the response.

Field	Description	Example
totalResults	Total number of elements in the database.	<totalResults></totalResults>
resultsPerPage	Number of items requested per page.	<resultsPerPage></resultsPerPage>
startIndex	The index of the first element returned.	<startIndex></startIndex>
nextPage	refURL to next page.	<nextPage></nextPage>
prevPage	refURL to previous page.	<prevPage></prevPage>
firstPage	refURL to first page.	<firstPage></firstPage>
lastPage	refURL to last page.	<lastPage></lastPage>
searchTerm	String value.	<searchTerm></searchTerm>
sortTerm	String value.	<sortTerm></sortTerm>

Important Notes

The following is a list of caveats and important notes about pagination.

- If you request a **startIndex** that is greater than total items, a full last page is returned.
- The **lastPage** should always return a full last page.

- The **firstPage** should always return a full first page (starting at 0).
- The **nextPage** is null if there is no **nextPage** (on last page).
- The **prevPage** is null if there is no **prevPage** (on first page).

Permissions Information

To facilitate making gadgets read-only for Supervisors, the APIs include permissions information, which indicates the operations that the user is allowed to perform.

The `<permissionInfo>` data is returned in the XML response when you perform a GET using the get/list operation. Here is an example:

```
<results>
  <pageInfo>...</pageInfo>
  <globalInfo>...</globalInfo>
  <permissionInfo>
    <canCreate>false</canCreate>
    <canUpdate>true</canUpdate>
    <canDelete>false</canDelete>
  </permissionInfo>
</results>

<congestionControl>
  <permissionInfo>
    <canUpdate>true</canUpdate>
    <canChangeDeploymentType>true</canChangeDeploymentType>
  </permissionInfo>
</congestionControl>
```

The `<canCreate>`, `<canUpdate>` and `<canDelete>` tags correspond to create, update and delete operations. You can only perform one of these operations if the corresponding tag is set to true.

If an API does not support a given operation, the corresponding permission is omitted. For instance, the Bulk Job API does not support updates, so it does not have a `<canUpdate>` tag.

If an API does not support any write operations, the `<permissionInfo>` tag is omitted entirely. This applies to the Active Directory Domain API, for example, because it cannot be modified in any way.

Search API

This section provides an overview of Search API, defines the search parameter, shows a search example, and outlines the default search values for existing configuration objects.

Overview

Search API has two parameters:

- **q**, where **q**=`<search_string>`
- **ignoreSearchErrors**, where **ignoreSearchErrors**=`[true|false]`

The **q** parameter is an optional search parameter taken by the various list API commands. It limits returned results to the configuration objects that match the search string.

You can perform a search on a predefined set of default fields for each configuration object. Typically, this is the **name** and **description** field, or the object's equivalent.

If **ignoreSearchErrors** is set to true, an invalid search string results in an empty list being returned rather than an API error. If this parameter is missing, it defaults to false. The setting **ignoreSearchErrors=true** should be specified if a client depends on list operations always succeeding for correct operation and/or has client-side search validation.

Search is subject to the following restrictions:

- Case-insensitive.
- String-only searches.
- Sql wildcards are not supported.
- "Contains" - the `<search_string>` will match any part of the default fields.
- The `<search_string>` is treated as a single string.
- An "or" search with a match on any of the default fields returning that record.

The search criteria are applied before the pagination parameters, so that pagination's **totalResults** value lists the total number of elements in the database that meet the search criteria.

Example

For example, a search for all the Call Types whose name or description contains "Supervisor" would be as follows:

```
https://<server>/unifiedconfig/config/calltype?q=supervisor
```

XML Returned

The following XML content is returned when the Search API is called.

```
<results>
  <pageInfo>
    <sortTerm>name</sortTerm>
    <searchTerm>supervisor</searchTerm>
    <firstPage>

[https://1.1.1.1/unifiedconfig/config/calltype?q=supervisor&sort=name%20asc&resultsPerPage=25]

    </firstPage>
    <lastPage>

[https://1.1.1.1/unifiedconfig/config/calltype?q=supervisor&sort=name%20asc&startIndex=0&
  resultsPerPage=25]
    </lastPage>
    <resultsPerPage>25</resultsPerPage>
    <startIndex>0</startIndex>
    <totalResults>1</totalResults>
  </pageInfo>
  <callTypes>
    <callType>
      <changeStamp>2</changeStamp>
      <refURL>[/unifiedconfig/config/calltype/5001]</refURL>
      <description>Used for Supervisor and Emergency Assist</description>
      <name>Assist</name>
    </callType>
  </callTypes>
</results>
```



Note

if you include Sort, a `<sortTerm>` tagged value is returned.

Default Search Values

The following table shows the default search values for existing configuration objects.

Configuration Object	Default Search Value
agent	<ul style="list-style-type: none"> • agentId • description • person.firstName • person.lastName • person.userName
agentDeskSettings	<ul style="list-style-type: none"> • name • description
agentTeam	<ul style="list-style-type: none"> • name • description
attribute	<ul style="list-style-type: none"> • name • description
bucketIntervals	<ul style="list-style-type: none"> • name
bulkJob	<ul style="list-style-type: none"> • description
callType	<ul style="list-style-type: none"> • name • description • id
dialedNumber	<ul style="list-style-type: none"> • dialedNumberString • description
expandedCallVariable	<ul style="list-style-type: none"> • name • description

networkVruScript	<ul style="list-style-type: none"> • name • description
precisionQueue	<ul style="list-style-type: none"> • name • description • id
reasonCode	<ul style="list-style-type: none"> • text • description
skillGroup	<ul style="list-style-type: none"> • name • description

Sort API

This section provides an overview of Sort API, defines the sort parameter, shows a sort example and the allowable sort attributes, and discusses error results.

Overview

You can sort **list** API results for each configuration object, in either an ascending or descending manner.

The parameter is: **sort=<attributeName> [asc|desc]**, where:

- **attributeName** is the name of the field as returned in the XML and is case-sensitive.
- **asc|desc** are optional and case-insensitive.
- **asc** stands for *ascending* sort, which is the default.
- **desc** stands for *descending* sort.



Note

Only the first sort argument on a GET query string is used to perform the sort operation. The others are ignored.

The sort option is applied after the search parameter and before the pagination parameters. The sort option can be specified by itself. The default sort field is the *name* field or equivalent.

Strings are returned in linguistic sort order. For example, *Alpha, abel, Beta, bagel* is sorted as:

- abel
- Alpha

- bagel
- Beta

Integer fields are sorted in integer order, not linguistic order. For example: 6, 12, 100 is sorted as 6, 12, 100.

Example

For example, to find all the CallTypes whose name or description contains *supervisor*, and to sort ascending by *name*:

```
https://<server>/unifiedconfig/config/calltype?q=supervisor&sort=name
```

And, to find all the Dialed Numbers and sort descending by *description*:

```
https://<server>/unifiedconfig/config/dialednumber?sort=description desc
```

XML Returned

The following XML content is returned when the Search API is called:

```
<results>
  <pageInfo>
    <sortTerm>name</sortTerm>
    <searchTerm>supervisor</searchTerm>
    <firstPage>

[https://1.1.1.1/unifiedconfig/config/calltype?q=supervisor&sort=name%20asc&resultsPerPage=25]

    </firstPage>
    <lastPage>

[https://1.1.1.1/unifiedconfig/config/calltype?q=supervisor&sort=name%20asc&startIndex=0&
  resultsPerPage=25]
    </lastPage>
    <resultsPerPage>25</resultsPerPage>
    <startIndex>0</startIndex>
    <totalResults>1</totalResults>
  </pageInfo>
  <callTypes>
    <callType>
      <changeStamp>2</changeStamp>
      <refURL>[/unifiedconfig/config/calltype/5001]</refURL>
      <description>Used for Supervisor and Emergency Assist</description>
      <name>Assist</name>
    </callType>
  </callTypes>
</results>
```



Note

For sort only commands, the <searchTerm> is not returned.

Allowable Sort Attributes

The following table shows the allowable sort attributes for existing configuration objects.

Configuration Object	Allowable Sort Attributes
agent	<ul style="list-style-type: none"> • agentId • description

	<ul style="list-style-type: none"> • supervisor • agentStateTrace • person.firstName • person.lastName • person.userName • person.loginEnabled
agentDeskSettings	<ul style="list-style-type: none"> • id • name • description • wrapupDataIncomingMode • wrapupDataOutgoingMode • remoteAgentType • logoutNonActivityTime • workModeTimer • supervisorAssistCallMethod • emergencyCallMethod • idleReasonRequired • logoutReasonRequired • autoAnswerEnabled
agentTeam	<ul style="list-style-type: none"> • id • name (default) • description
attribute	<ul style="list-style-type: none"> • id • name • dataType • defaultValue • description
bucketIntervals	<ul style="list-style-type: none"> • id • name (default)

	<ul style="list-style-type: none"> • upperBound1-9
bulkJob	<ul style="list-style-type: none"> • id • description • jobType • jobState • jobHostName • createDateTime • startDateTime • endDateTime
callType	<ul style="list-style-type: none"> • name (default) • description • id • serviceLevelThreshold • serviceLevelType
dialedNumber	<ul style="list-style-type: none"> • id • dialedNumberString (default) • description
expandedCallVariable	<ul style="list-style-type: none"> • id • name (default) • description • maximumLength • maximumArraySize • eccArray • enabled • persistent • ciscoProvided
networkVruScript	<ul style="list-style-type: none"> • id • name (default)

	<ul style="list-style-type: none"> • description • vruScriptName • timeout • configParam • interruptible
precision Queue	<ul style="list-style-type: none"> • name • description • id
reasonCode	<ul style="list-style-type: none"> • id • text (default) • description • code
skillGroup	<ul style="list-style-type: none"> • id • name • description • serviceLevelThreshold • serviceLevelType • peripheralNumber

Error Results

Specifying an invalid sort field or sort option (**asc|desc**) or too many parameters in the sort request results in an `apiError` being returned with `ErrorType` set to **invalidInput.badSortField**.

For example, the sort parameter: **sort=name asc extra** results in the following `apiError`:

```
<apiErrors>
  <apiError>
    <errorType>invalidInput.badSortField</ErrorType>
    <errorData>name asc extra</ErrorData>
    <errorMessage> .... </ErrorMessage>
  </apiError>
</apiErrors>
```

Asynchronous API

Async calls can offer greater efficiency over synchronous calls and you may want to use async when the system is busy.

Whereas synchronous API calls are blocking calls and do not return until the change has been completed, or there has been an error, with async the response to the API call is returned immediately, with a polling URL, while the request continues to be processed. In heavier load conditions it can be more efficient to submit multiple async calls and periodically check their status than to wait for each call to complete before submitting the next one.

By default, if an API call is made from the UI it has five seconds to get a response from the API before the UI framework times out on that request, even if the request is still being processed. With synchronous calls, in a case where the request takes over five seconds to complete, a request timed-out error is returned to the user, even though the request may have completed successfully. With async, the UI can poll the status of the request until it is completed and can display an appropriate processing symbol, such as an hour glass, until the request either completes, truly does error out, or reaches a timeout that is deemed too long.

This section explains how to make an Asynchronous (Async) call and outlines the expected responses. It describes the three supported operations for this API: **create**, **update**, and **delete**. The section also provides four use cases, and explains the exceptions returned and the conditions under which they are returned.

The examples shown describe how to use the Async feature to create a Call Type.

create

URL:	https://<server>/unifiedconfig/config/calltype?async=true
HTTP Method:	POST
Response:	<p>If a request is successfully put on the queue for processing—that is, if it has passed the validation check before getting on queue—the result is the HTTP Response 202 Accepted with the following Location URL in the header, for polling the status of the request:</p> <p>URL: https://<server>/unifiedconfig/config/asyncrequeststatus/<id></p> <p>And, the following XML content is returned:</p> <pre><asyncResult> <progress>IN_QUEUE</progress> </asyncResult></pre>
Exceptions	See Exceptions , on page 24.

update

URL:	https://<server>/unifiedconfig/config/calltype/<ID>?async=true
HTTP Method:	PUT

Response:	<p>If a request is successfully put on the queue for processing—that is, if it has passed the validation check before getting on queue—the result is the HTTP Response 202 Accepted with the following Location URL in the header, for polling the status of the request:</p> <p>URL: <code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code></p> <p>And, the following XML content is returned:</p> <pre><asyncResult> <progress>IN_QUEUE</progress> </asyncResult></pre>
Exceptions	See Exceptions, on page 24 .

delete

URL:	<code>https://<server>/unifiedconfig/config/calltype/<ID>?async=true</code>
HTTP Method:	DELETE
Response:	<p>If a request is successfully put on the queue for processing—that is, if it has passed the validation check before getting on queue—the result is the HTTP Response 202 Accepted with the following Location URL in the header, for polling the status of the request:</p> <p>URL: <code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code></p> <p>And, the following XML content is returned:</p> <pre><asyncResult> <progress>IN_QUEUE</progress> </asyncResult></pre>
Exceptions	See Exceptions, on page 24 .

Values

The following table shows the values for <progress> in the returned XML content:

Value	Description
IN_QUEUE	The request passed validation and capacity checks and was put on the queue.
IN_PROGRESS	The request has been taken off the queue and is being processed.

Use Cases

For further explanation, this section provides four use cases.

Create (success)

1) User sends an Asynchronous request to create a Call Type.	Example request: <code>https://<server>/unifiedconfig/config/calltype?async=true</code>
2) System validates the request and puts it in queue and returns the following:	<ul style="list-style-type: none"> • 202 Accepted status with progress of IN_QUEUE; and with • A URL in the location header that can be polled for further status information. • ◦ Example URL for polling: <code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
3) User polls the status using the provided URL:	<code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
4) Depending on timing, as this is a very short state, the system may return the following status to indicate that the request has been taken out of queue and is being processed:	202 Accepted status with progress of IN_PROGRESS.
5) User polls the status again using the previously provided URL:	<code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
6) System returns the following status to indicate the Call Type was created successfully:	201 Created status with a refUrl of <code>https://<server>/unifiedconfig/config/calltype/<id></code>

Update (success)

1) User sends an Asynchronous request to update an existing Call Type object.	Example request: <code>https://<server>/unifiedconfig/config/calltype/<id>?async=true</code>
2) System validates the request and puts it in queue and returns the following:	<ul style="list-style-type: none"> • 202 Accepted status with progress of IN_QUEUE; and with • A URL in the location header that can be polled for further status information. • ◦ Example URL for polling: <code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
3) User polls the status using the provided URL:	<code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
4) System returns a 200 OK status to indicate the Call Type was updated successfully.	

Delete (success)

1) User sends an Asynchronous request to delete an existing Call Type object.	Example request: <code>https://<server>/unifiedconfig/config/calltype/<id>?async=true</code>
2) System validates the request and puts it in queue and returns the following:	<ul style="list-style-type: none"> • 202 Accepted status with progress of IN_QUEUE; and with • A URL in the location header that can be polled for further status information. <ul style="list-style-type: none"> ◦ Example URL for polling: <code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
3) User polls the status using the provided URL:	<code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
4) System returns a 200 OK status to indicate the Call Type was deleted successfully.	

Create (failure)

1) User sends an Asynchronous request to create a Call Type.	Example request: <code>https://<server>/unifiedconfig/config/calltype?async=true</code>
2) System validates the request and puts it in queue and returns the following:	<ul style="list-style-type: none"> • 202 Accepted status with progress of IN_QUEUE; and with • A URL in the location header that can be polled for further status information. <ul style="list-style-type: none"> ◦ Example URL for polling: <code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
3) User polls the status using the provided URL:	<code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
4) Depending on timing, as this is a very short state, the system may return the following status to indicate that the request has been taken out of queue and is being processed:	202 Accepted status with progress of IN_PROGRESS.
5) User polls the status again using the previously provided URL:	<code>https://<server>/unifiedconfig/config/asyncrequeststatus/<id></code>
6) System returns the following status to indicate that the system capacity has been exceeded for Call Type:	400 Bad Request , with error text.

Exceptions

This section explains the exceptions returned and the conditions under which they are returned:

Exceptions:

- Tasks that cannot be put on the queue due to max capacity return an HTTP status code of **503**, with an API error indicating the queue is full.
- If a task reaches its max time in queue of 30 seconds, it is removed from the queue. On the next poll for the status of this task, an HTTP status code of **503** is returned, with an API error indicating timed out.



PART **II**

The Cisco Packaged Contact Center Enterprise Developers Guide APIs

- [Active Directory Domain API, page 27](#)
- [Agent API, page 29](#)
- [Agent Desk Settings API, page 43](#)
- [Agent State Trace API, page 49](#)
- [Agent Team API, page 53](#)
- [Attribute API, page 59](#)
- [Bucket Intervals, page 65](#)
- [Bulk Job API, page 71](#)
- [Call Type API, page 91](#)
- [Congestion Control API, page 97](#)
- [Deployment API, page 101](#)
- [Deployment Type Info API, page 103](#)
- [Dialed Number API, page 111](#)
- [Expanded Call Variable APIs, page 119](#)
- [Media Routing Domain API, page 125](#)
- [Network VRU Script APIs, page 127](#)

- [Precision Queue API](#), page 133
- [Reason Code API](#), page 147
- [Serviceability API](#), page 151
- [Skill Group API](#), page 161



CHAPTER 2

Active Directory Domain API

You can use the Active Directory Domain API to list the Active Directory Domains currently defined in your call center environment.

- [Active Directory Domain API Commands, page 27](#)

Active Directory Domain API Commands

This section explains the supported API operation for Active Directory Domain.

list

Retrieves a list of Active Directory Domains.



Note

This API only supports sorting in alphabetic order.

URL:	https://<server>/unifiedconfig/config/activedirectorydomain Note Note: The Active Directory Domain API does not require a user to authenticate, so anyone can do a GET on this API.
HTTP Method:	GET
Example XML Response:	<pre><results> <activeDirectoryDomains> <activeDirectoryDomain> <name>boston.com</name> </activeDirectoryDomain> <activeDirectoryDomain> <name>cisco.com</name> </activeDirectoryDomain> </activeDirectoryDomains> </results></pre> <p>See HTTP Responses, on page 7.</p>



Agent API

You can use the Agent API to list the Agents currently defined in the database, define new Agents, and view, edit, or delete records of existing Agents.

- [Agent API Commands, page 29](#)

Agent API Commands

This section explains the five supported API operations for Agent and outlines the parameters.

API Parameters

The following table shows the parameters for Agent API.

Parameter Name	Description
agentId	The peripheral number.
description	The description of this desk setting.
agentStateTrace	If true, turn on Agent State Tracing for the agent. A maximum of 100 agents can have this flag set to true.
changeStamp	The version of the agent object. Initially set by the database during a create . Must be passed back for update operations.
AgentDeskSettings.refURL	refURL to the agentDeskSettings object.
person.firstName	First name of the agent/person.
person.lastName	Last name of the agent/person.
person.userName	User name of the agent/person.

Parameter Name	Description
person.password	Password for the agent/person. Note This is not returned in read operations.
person.loginEnabled	Indicates whether an agent/person can log in.
supervisor	Indicates whether the agent is marked as supervisor.
supervisorUserInfo.userName	The Active Directory userName for this supervisor.
supervisorUserInfo.domainName	The Active Directory Domain name for this supervisor. If empty, it uses the default domain name.
agentAttribute.attributeValue	The value of the attribute for the agent.
agentAttribute.description	Per agent attribute description.
agentAttribute.attribute.refURL	The refURL to the attribute object.
agentAttribute.attribute.name	Name of the attribute.
agentAttribute.attribute.dataType	Type of attribute.
agentAttribute.attribute.description	Attribute description.
skillGroup.refURL	The refURL to the skill group.
skillGroup.name	Name of the skill group.
defaultSkillGroup.refURL	The refURL to the default skill group.
defaultSkillGroup.name	Name of the default skill group.
agentTeam.refURL	The refURL to the agent team.
agentTeam.name	Name of the agent team.
supervisorTeam.refURL	The refURL to the supervisor's team.
supervisorTeam.name	Name of the supervisor's team.

create

Creates an Agent record and stores the data in the database.

URL:	<code>https://<server>/unifiedconfig/config/agent</code>
-------------	--

HTTP Method:	POST
Input/Output Format:	XML
Parameters:	See API Parameters , on page 29
Example Response:	<pre> <agent> <agentId>8006</agentId> <agentStateTrace>>false</agentStateTrace> <description>an agent</description> <person> <firstName>Agent2</firstName> <lastName>Agent2</lastName> <loginEnabled>>true</loginEnabled> <userName>Agent2</userName> <password>mypassword</password> </person> <agentDeskSettings> <refURL>/unifiedconfig/config/agentdesksetting/5434</refURL> </agentDeskSettings> <supervisor>>true</supervisor> <supervisorUserInfo> <userName>boston</userName> <domainName>boston.com</domainName> </supervisorUserInfo> <agentAttributes> <agentAttribute> <attribute> <refURL>/unifiedconfig/config/attribute/5004</refURL> <name>Sales</name> <dataType>4</dataType> <description>Sales proficiency</description> <attributeValue>8</attributeValue> <description>masters certification</description> </agentAttribute> </agentAttributes> <skillGroups> <skillGroup><refURL>/unifiedconfig/config/skillgroup/18434</refURL> </skillGroup> <skillGroup><refURL>/unifiedconfig/config/skillgroup/18435</refURL> </skillGroup> </skillGroups> <defaultSkillGroup><refURL>/unifiedconfig/config/skillgroup/18434</refURL> </defaultSkillGroup> <agentTeam><refURL>/unifiedconfig/config/agentteam/5003</refURL></agentTeam> <supervisorTeams> <supervisorTeam><refURL>/unifiedconfig/config/agentteam/5003</refURL> </supervisorTeam> <supervisorTeam><refURL>/unifiedconfig/config/agentteam/5006</refURL> </supervisorTeam> </supervisorTeams> </agent> </pre>
Response:	In the Response, the Location header has a URL to the newly created Agent, if successful. See also HTTP Responses , on page 7.

Operation Validation:	
----------------------------------	--

agentId

- Optional field.
- Max: 11-digit number.
- Leading zeros are allowed.
- Auto-generated if not supplied.

description

- Optional field.
- No restriction of characters.
- Max length of 255 bytes allowed.
- For details on valid characters for this field, see [Internationalization](#), on page 9.

agentStateTrace

- Optional field.
- True or false; default value is false.
- Capacity limit of 100 Agents with this parameter set to true.

agentDeskSettings.refURL

- Optional field.
- Valid AgentDeskSettings URL.
- Null is also valid.

changeStamp

- Optional field.
- Integers only: start with 0.

person.userName

- Required field.
- Max length of 32 bytes allowed.
- Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric.
- Name must be unique.
- Does not allow internationalized characters.

person.firstName

- Required field.
- No restriction of characters.

- Max length of 32 bytes.
- For details on valid characters for this field, see [Internationalization](#), on page 9.

person.lastName

- Required field.
- No restriction of characters.
- Max length of 32 bytes.
- For details on valid characters for this field, see [Internationalization](#), on page 9.

person.password

- Optional field.
- No restriction of characters.

person.loginEnabled

- Optional field.
- True or false; default value is true.

supervisor

- Optional field.
- If present and set as true, supervisorInfo element must exist.
- If set as false, supervisorInfo is ignored.

userName

- Required field if the agent is set to be a supervisor.
- Max: 64 bytes.
- Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric.
- User must exist in the Active Directory.

domainName

- Optional field: if missing, uses the default domain name.
- Max: 64 bytes.
- Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric.

agentAttributes

- Optional field.
- Zero or more attributes can be specified.
- The **attribute.refURL** must refer to valid, non-deleted attributes. It must be specified.
- The **agentAttribute.attributeValue** must be valid for the attribute type (for boolean: true or false; for proficiency: integer between 1 and 10 inclusive).
- No more than 50 attribute values per Agent.

skillGroups

- Optional field.
- Zero or more Skill Groups can be specified.
- The **skillGroup.refURL** must refer to a valid, non-deleted Skill Group. If a Skill Group is specified, the refURL must be specified.

defaultSkillGroup

- Optional field.
- Only one **defaultSkillGroup** can be specified.
- If the **defaultSkillGroup** tag is specified, it must reference a Skill Group that is listed in the **skillGroups**. That is, it must reference a Skill Group that the Agent is a member of.

agentTeam

- Optional field.
- An agent can only belong to one team, at most.
- If the **agentTeam** tag is specified, the **agentTeam.refURL** tag must be specified and refer to a valid team.
- A team can contain up to 50 agents.

supervisorTeam

- The Agent must be a Supervisor.
- The **supervisorTeam.refURL** tag must be valid and refer to a valid team.
- A Supervisor can supervise from 0-20 teams.
- A team can have 0-10 Supervisors.
- A Supervisor can be both a member of and a Supervisor of the same team.

delete

Deletes one Agent record and the associated Person object from the database.

The **delete** operation automatically deletes any Skill, Attribute, Team, or Supervisor reference to the Agent.

**Note**

The operation only *marks* the record for deletion, it does not permanently delete it.

URL:	https://<server>/unifiedconfig/config/agent/<id>
HTTP Method:	DELETE
Response:	See HTTP Responses , on page 7.
Operation Validation:	<p>You cannot delete an Agent that is referenced by a Script.</p> <p>You cannot delete an agent that is a supervisor assigned to an Agent Team.</p> <p>When you try to delete an Agent who is assigned as Supervisor, the associated record created in UserGroup is deleted. In addition, other items are cleaned up, such as the record in User_Supervisor_Map.</p>

list

Retrieves a list of agents.

URL:	https://<server>/unifiedconfig/config/agent
HTTP Method:	GET

<p>Example XML Response:</p>	<pre><results> <pageInfo> ... (see pagination data) </pageInfo> <agents> <agent>...</agent> <agent>...</agent> </agents> </results></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10.</p> <p>Also, the example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See also HTTP Responses, on page 7.</p>
<p>Summary List</p>	<p>By default, the list command returns a list of Agents with all the agent data that is returned by the GET command documented below. However if Agents contain a large number of skillgroups and attributes, this can result in a large amount of data being returned. Therefore the summary=true query option returns a much smaller amount of data per agent. It just returns the basic agent data and the person data. An example is performing a GET on <a href="https://<server>:<serverport>/unifiedconfig/config/agent2?summary=true">https://<server>:<serverport>/unifiedconfig/config/agent2?summary=true.</p> <pre><results> <pageInfo> ... (see pagination data) </pageInfo> <agents> <agent xsi:type="agentSummary"> <changeStamp>7</changeStamp> <refURL>/unifiedconfig/config/agent/10884</refURL> <agentId>4294305</agentId> <agentStateTrace>>false</agentStateTrace> <agentTeam> <refURL>/unifiedconfig/config/agentteam/5005</refURL> <name>monsters</name> </agentTeam> <description>Here is a descr</description> <person> <firstName>as</firstName> <lastName>al</lastName> <loginEnabled>>true</loginEnabled> <userName>aaa</userName> </person> <supervisor>>false</supervisor> </agent> <agent>...</agent> </agents> </results></pre> <p>Note that the "<agent ..>" tag has the attribute: xsi:type="agentSummary" that indicates it contains a agent summary object. For non-summary lists, the value would be: xsi:type="agent".</p>

Supervisor Search

The **supervisor** search parameter allows you to specify a search to find Agents that are (or are *not*) supervisors.

For example:

- **q=supervisor:true** Returns all agents who are supervisors.
- **q=supervisor:false** Returns all agents who are *not* supervisors.

Advanced Search

In addition to the supervisor search, you can perform the following advanced searches:

- **attributes: (attr1 & attr2 & attr3, ...)**
- **skillgroups: (skill1 & skill2 & skill3,...)**
- **team: (team1|team2|team3, ...)**

The **attributes** search returns *all* Agents that have *all* the specified attributes. Up to ten attributes can be specified. The attribute names are fully matched.

The **skillgroups** search returns *all* agents that have *all* the specified skillgroups. Up to ten skillgroups can be specified. The skillgroup names are fully matched.

The **team** search returns *all* agents who belong to *any* of the specified teams. Up to ten team names can be specified. The team name is fully matched.

The following restrictions apply:

- Searches for attribute names, skillgroup names, and team names are case insensitive.
- There must be a space separator between any of the special search terms.
- The **&** separator within the **attributes:()** and **skillgroups:()** search must be URL-encoded to %26 before being sent to the server, otherwise the server will interpret it as an http parameter separator.
- The search terms can be separated in any order, such as:

q=test description attributes:(a1 & a2) skillgroups:(sk1 & sk5) team:(sales | support) supervisor:false

- The above must be sent as: **q=test description attributes:(a1 %26 a2) skillgroups:(sk1 %26 sk5) team:(sales | support) supervisor:false**

Or:

q=team:(sales) attributes:(a1 & a2) test description

- The above must be sent as: **q=team:(sales) attributes:(a1 %26 a2) test description**

get

Returns one Agent record from the database.

URL:	https://<server>/unifiedconfig/config/agent/<id>
HTTP Method:	GET

**Example
XML
Response:**

```

<agent>
  <changeStamp>2877</changeStamp>
  <refURL>/unifiedconfig/config/agent/5017</refURL>
  <agentId>8006</agentId>
  <agentStateTrace>>false</agentStateTrace>
  <description>an agent</description>
  <person>
    <firstName>Agent2</firstName>
    <lastName>Agent2</lastName>
    <loginEnabled>>true</loginEnabled>
    <userName>Agent2</userName>
    <password>mypassword</password>
  </person>
  <agentDeskSettings>
    <name>test2</name>
    <refURL>/unifiedconfig/config/agentdesksetting/5434</refURL>
  <supervisor>>true</supervisor>
</agentDeskSettings>
<supervisorUserInfo>
  <userName>boston</userName>
  <domainName>boston.com</domainName>
</supervisorUserInfo>
<agentAttributes>
  <agentAttribute>
    <attribute>
      <refURL>/unifiedconfig/config/attribute/5004</refURL>

      <name>Sales</name>
      <dataType>4</dataType>
      <description>Sales proficiency</description>
    </attribute>
    <attributeValue>8</attributeValue>
    <description>masters certification</description>
  </agentAttribute>
</agentAttributes>
<skillGroups>
  <skillGroup>
    <refURL>/unifiedconfig/config/skillgroup/5229</refURL>
    <name>Support</name>
  </skillGroup>
</skillGroups>

<defaultSkillGroup>
  <refURL>/unifiedconfig/config/skillgroup/5229</refURL>
  <name>Support</name>
</defaultSkillGroup>

  <agentTeam>
    <refURL>/unifiedconfig/config/agentteam/5003</refURL>
    <name>theTeam</name>
  </agentTeam>
  <supervisorTeams>
    <supervisorTeam>
      <refURL>/unifiedconfig/config/agentteam/5003</refURL>
      <name>theTeam</name>
    </supervisorTeam>
    <supervisorTeam>
      <refURL>/unifiedconfig/config/agentteam/5006</refURL>
      <name>theBTeam</name>
    </supervisorTeam>
  </supervisorTeams>
</agent>

```

See [HTTP Responses](#), on page 7.

**Example
XML
Response, if**

```

<agent>
  <changeStamp>2877</changeStamp>
  <refURL>/unifiedconfig/config/agent/5017</refURL>
  <agentId>8006</agentId>
  <agentStateTrace>>false</agentStateTrace>

```

Supervisor is False:

```

<description>an agent</description>
<person>
  <firstName>Agent2</firstName>
  <lastName>Agent2</lastName>
  <loginEnabled>true</loginEnabled>
  <userName>Agent2</userName>
  <password>mypassword</password>
</person>
<agentDeskSettings>
  <name>test2</name>
  <refURL>/unifiedconfig/config/agentdesksetting/5434</refURL>
</agentDeskSettings>
<supervisor>>false</supervisor>
<agentAttributes>
  <agentAttribute>
    <attribute>
      <refURL>/unifiedconfig/config/attribute/5004</refURL>

      <name>Sales</name>
      <dataType>4</dataType>
      <description>Sales proficiency</description>
    </attribute>
    <attributeValue>8</attributeValue>
    <description>masters certification</description>
  </agentAttribute>
</agentAttributes>
</agent>

```

update

Updates one Agent record in the database.

URL:	https://<server>/unifiedconfig/config/agent/<id>
HTTP Method:	PUT
Input/Output Format:	xml

Example XML Request Payload:	<pre> <agent> <changeStamp>2877</changeStamp> <refURL>/unifiedconfig/config/agent/5017</refURL> <agentId>8006</agentId> <agentStateTrace>>false</agentStateTrace> <description>an agent</description> <person> <firstName>Agent2</firstName> <lastName>Agent2</lastName> <loginEnabled>>true</loginEnabled> <userName>Agent2</userName> <password>mypassword</password> </person> <agentDeskSettings> <name>test2</name> <refURL>/unifiedconfig/config/agentdesksetting/5434 </refURL> <supervisor>>true</supervisor> <supervisorUserInfo> <userName>boston</userName> <domainName>boston.com</domainName> </supervisorUserInfo> </agentDeskSettings> <agentAttributes> <agentAttribute> <attribute> <refURL>/unifiedconfig/config/attribute/5004</refURL> <name>Sales</name> <dataType>4</dataType> <description>Sales proficiency</description> </attribute> <attributeValue>8</attributeValue> <description>masters certification</description> </agentAttribute> </agentAttributes> </agent> </pre>
Parameters:	See API Parameters , on page 29.
Response:	See HTTP Responses , on page 7.
Operation Validation:	Restrictions: <ul style="list-style-type: none"> • All of the restrictions from the create operation apply. • Also: you must specify valid Change Stamps for the Agent.
Attribute Update Behavior:	<ul style="list-style-type: none"> • If you specify a list of Agent attributes, the list replaces the current list of Agent attributes for the specified Agent. • If you specify an empty list of Agent attributes via the <agentAttributes/> tag or equivalent, any current attributes are removed from the Agent. • If you do not specify an <agentAttributes/> tag, the current settings remain.
Supervisor Update Behavior:	If you change the supervisor from true to false, you must make sure that supervisor is not referenced by an Agent Team.

SkillGroup Update Behavior:	<ul style="list-style-type: none"> • If you specify a list of Skill Groups, it replaces the current list of Skill Groups for the specified Agent. • If you specify an empty list of Skill Groups via the <skillGroups/> tag or equivalent, the Agent is removed from all Skill Groups. • If you do not specify a skillGroups tag, the current Skill Groups settings remain.
DefaultSkillGroup update Behavior:	If you change the list of Skill Groups that the Agent is a member of, you must also update the defaultSkillGroup to either be Null or point to one of the new Skill Groups.
AgentTeam Update Behavior:	<ul style="list-style-type: none"> • If you specify an Agent Team, the Agent is removed from the current team (if they are already in one) and added to the specified team. • If you specify an empty team via the <agentTeam/> tag or equivalent, the Agent is removed from the current team. • If you do not specify an agentTeam tag, the current Agent Team settings remain.
SupervisorTeam Update Behavior:	<ul style="list-style-type: none"> • You must be a Supervisor to specify Supervisor Teams. • If you specify a list of Supervisor Teams, the list replaces the current list of Supervisor Teams for the Supervisor. • All supervisorTeam.refURLs must be valid refURLs. • If you specify an empty list of Supervisor Teams via the supervisorTeams tag or equivalent, the Supervisor is removed as a Supervisor from all teams they currently supervise. • A Supervisor can be both a member of and a Supervisor of the same team. • If you demote Supervisors to non-supervisors, they are no longer Supervisors for any of their previous teams. In a case where a Supervisor already supervised teams, you must delete that Supervisor's supervisorTeam settings when demoting the supervisor by specifying the empty <supervisorTeams/> tag.

Asynchronous API

See section on [Asynchronous API](#), on page 20.



Note

For Agent API, the Asynchronous feature is supported only for the create, update, and delete operations.



Agent Desk Settings API

You can use the Agent Desk Settings API to list the Agent Desk Settings currently defined in the database, define new Agent Desk Settings, and view, edit, or delete records of existing Agent Desk Settings.

- [Agent Desk Settings API Commands, page 43](#)

Agent Desk Settings API Commands

This section explains the five supported API operations for Agent Desk Settings and their parameters.

API Parameters

The following table shows the parameters for Agent Desk Settings API:

Parameter Name	Description
enterpriseName	The Enterprise name of the Agent Desk Setting.
description	The description of the Agent Desk Setting.
wrapupDataIncomingMode	Indicates whether the Agent is allowed or required to enter wrap-up data after an inbound call.
wrapupDataOutgoingMode	Indicates whether the Agent is allowed or required to enter wrap-up data after an outbound call.
remoteAgentType	Determines how mobile Agents using this dial plan are handled.
logoutNonActivityTime	Number of seconds of non-activity at the desktop after which the software automatically logs out the Agent.
workModeTimer	Wrap-up time: specifies the auto wrap-up time out. Value must be between 1 and 7200 (default is 7200).

Parameter Name	Description
supervisorAssistCallMethod	Indicates whether Unified CCE creates a consultative call or a blind conference call for the supervisor assistance request.
emergencyCallMethod	Indicates whether Unified CCE creates a consultative call or a blind conference call for an emergency call request.
idleReasonRequired	Indicates whether the Agent must enter a reason before entering the Idle state.
logoutReasonRequired	Indicates whether the Agent must enter a reason before logging out.
AutoAnswerEnabled	Indicates if Is auto answer is enabled.

create

Creates one Agent Desk Setting and stores it in the database.

URL:	https://<server>/unifiedconfig/config/agentdesksetting
HTTP Method:	POST
Input/Output Format:	xml
Parameters:	See API Parameters , on page 43.
Example XML Request Payload:	<pre><agentDeskSetting> <name>test</name> <description>test agent desk setting</description> <wrapupDataIncomingMode>1</wrapupDataIncomingMode> <wrapupDataOutgoingMode>1</wrapupDataOutgoingMode> <logoutNonActivityTime>30</logoutNonActivityTime> <logoutReasonRequired>true</logoutReasonRequired> <idleReasonRequired>false</idleReasonRequired> <autoAnswerEnabled>true</autoAnswerEnabled> <changeStamp>0</changeStamp> </agentDeskSetting></pre>
Response:	In the Response, the Location header has a URL to the newly created Agent Desk Setting, if successful. See also HTTP Responses , on page 7.
Operation Validation:	<p>name</p> <ul style="list-style-type: none"> • Required field. • Max length of 32 bytes allowed. • Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric.

- Name must be unique.
- Does not allow internationalized characters.

description

- Optional field.
- No restriction of characters.
- Max length of 255 bytes allowed.
- For details on valid characters for this field, see [Internationalization](#), on page 9.

wrapupDataIncomingMode

- Required field.
- Value must be 0, 1, 2 (default to 1).

wrapupDataOutgoingMode

- Required field.
- Value must be 0, 1, 2 (default to 1).

remoteAgentType

- Optional field.
- Value must be 0, 1, 2, 3 (default to 0).

logoutInActivityTime

- Optional field.
- Value must be between 10 and 7200 (default to NULL).

workModeTimer

- Optional field.
- Value must be between 1 and 7200 (default to 7200).

supervisorAssistCallMethod

- Optional field.
- Value must be 0, 1 (default to 0).

emergencyCallMethod

- Optional field.
- Value must be 0, 1 (default to 0).

idleReasonRequired

	<ul style="list-style-type: none"> • Optional field. • Boolean: true/false (default is false). <p>logoutReasonRequired</p> <ul style="list-style-type: none"> • Optional field. • Boolean: true/false (default is false). <p>AutoAnswerEnabled</p> <ul style="list-style-type: none"> • Optional field. • Boolean: true/false (default is false). <p>changeStamp</p> <ul style="list-style-type: none"> • Optional field. • Integers only: start with 0.
--	---

delete

Permanently deletes one Agent Desk Setting from the database.

URL:	https://<server>/unifiedconfig/config/agentdesksetting/<id>
HTTP Method:	DELETE
Response:	See HTTP Responses , on page 7.
Operation Validation:	You cannot delete any Agent Desk Setting that is: <ul style="list-style-type: none"> • Referenced by a Peripheral. • Referenced by an Agent.

list

Retrieves a list of Agent Desk Settings.

URL:	https://<server>/unifiedconfig/config/agentdesksetting
HTTP Method:	GET

Example XML Response:	<pre><results> <pageInfo> ... (see pagination data) </pageInfo> <agentDeskSettings> <agentDeskSetting>...</agentDeskSetting> <agentDeskSetting>...</agentDeskSetting> </agentDeskSettings> </results></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10.</p> <p>Note Also, the example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See also HTTP Responses, on page 7.</p>
------------------------------	--

get

Returns one Agent Desk Setting from the database.

URL:	https://<server>/unifiedconfig/config/agentdesksetting/<id>
HTTP Method:	GET
Example XML Response:	<pre><agentDeskSetting> <refURL>http://<server>/unifiedconfig/config/agentDeskSetting/(id)</refURL> <name>test</name> <description>test agent desk setting</description> <wrapupDataIncomingMode>1</wrapupDataIncomingMode> <wrapupDataOutgoingMode>1</wrapupDataOutgoingMode> <logoutNonActivityTime>30</logoutNonActivityTime> <logoutReasonRequired>true</logoutReasonRequired> <idleReasonRequired>false</idleReasonRequired> <autoAnswerEnabled>true</autoAnswerEnabled> <changeStamp>0</changeStamp> </agentDeskSetting></pre> <p>See HTTP Responses, on page 7.</p>

update

Updates one Agent Desk Setting in the database.

URL:	https://<server>/unifiedconfig/config/agentdesksetting/<id>
HTTP Method:	PUT
Input/Output Format:	xml

Example XML Request Payload:	<pre><agentDeskSetting> <name>test</name> <description>test agent desk setting</description> <wrapupDataIncomingMode>1</wrapupDataIncomingMode> <wrapupDataOutgoingMode>1</wrapupDataOutgoingMode> <logoutNonActivityTime>30</logoutNonActivityTime> <logoutReasonRequired>true</logoutReasonRequired> <idleReasonRequired>false</idleReasonRequired> <autoAnswerEnabled>true</autoAnswerEnabled> <changeStamp>0</changeStamp> </agentDeskSetting></pre>
Parameters:	See API Parameters , on page 43.
Response:	See HTTP Responses , on page 7.
Operation Validation:	All of the restrictions from the Create operation apply.

Asynchronous API

See section on [Asynchronous API](#), on page 20.



Note

For Agent Desk Settings API, the Asynchronous feature is supported only for the create, update, and delete operations.



Agent State Trace API

You can use the Agent State Trace API to view and edit Agent State Trace.

- [Agent State Trace API Commands](#), page 49

Agent State Trace API Commands

This section explains the two supported API operations for Agent State Trace and their parameters.

API Parameters

See [API Parameters](#), on page 29.

get

Returns a list of Agents whose Agent State Trace is turned on from the database.

URL:	https://<server>/unifiedconfig/config/agentstatetrace
HTTP Method:	GET

Example XML Response:	<pre> <agentstatetrace> <refURL>/unifiedconfig/config/agentstatetrace</refURL> <agents> <agent xsi:type="agentSummary"> <refURL>/unifiedconfig/config/agent/10884</refURL> <agentId>4294305</agentId> <agentStateTrace>true</agentStateTrace> <description>Here is a descr</description> <person> <firstName>as</firstName> <lastName>a1</lastName> <loginEnabled>true</loginEnabled> <userName>aaa</userName> </person> <supervisor>>false</supervisor> </agent> </agents> </agentstatetrace> </pre> <p>The example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See HTTP Responses, on page 7.</p>
------------------------------	--

update

Updates the Agent State Trace in the database.

Note: Turn on the Agent State Trace for the Agents, but turn off the trace for the rest of Agents who are not specified in the request. To turn off all the Agent State Trace, simply pass in an empty list.

URL:	https://<server>/unifiedconfig/config/agentstatetrace
HTTP Method:	PUT
Input/Output Format:	xml
Example XML Request Payload:	<pre> <agentstatetrace> <agents> <agent><refURL>/unifiedconfig/config/agent/7708</refURL></agent> <agent><refURL>/unifiedconfig/config/agent/7709</refURL></agent> <agent><refURL>/unifiedconfig/config/agent/7372</refURL></agent> <agent><refURL>/unifiedconfig/config/agent/7711</refURL></agent> </agents> </agentstatetrace> </pre>
Parameters:	See API Parameters, on page 29 .
Response:	See HTTP Responses, on page 7 .
Operation Validation:	<p>Agent.refURL</p> <ul style="list-style-type: none"> • Must refer to a valid agent.

	<ul style="list-style-type: none">• The format of the URL must be valid.• Cannot contain any invalid characters.
Capacity Checking:	The maximum number of Agents with Agent State Trace on is 100.



CHAPTER

6

Agent Team API

You can use the Agent Team API to list the Agent Teams currently defined in the database, define new Agent Teams, and view, edit, or delete records of existing Agent Teams.

- [Agent Team API Commands, page 53](#)

Agent Team API Commands

This section explains the five supported API operations for Agent Team and outlines the parameters.

API Parameters

The following table shows the parameters for Agent Team API:

Parameter Name	Description
name	The name of the Agent Team. This name must be unique among all Agent Teams in the system.
dialedNumber refURL	A reference to an internal Dialed Number for the Agent Team.
description	Additional information about the Agent Team.
agent refURL	A reference to an Agent.
supervisor refURL	A reference to a Supervisor.

create

Creates an Agent Team record and stores the data in the database.

URL:	<code>https://<server>/unifiedconfig/config/agentteam</code>
-------------	--

HTTP Method:	POST
Input/Output Format:	xml
Parameters:	See API Parameters , on page 53.
Example XML Request Payload:	<pre> <agentTeam> <name>team1</name> <dialNumber> <refURL>[/unifiedconfig/config/dialednumber/ (id)]</refURL> </dialNumber> <description>test agent team1</description> <agents> <agent> <refURL>[/unifiedconfig/config/agent/ (id_1)]</refURL> </agent> <agent> <refURL>[/unifiedconfig/config/agent/ (id_2)]</refURL> </agent> </agents> <supervisors> <supervisor> <refURL>[/unifiedconfig/config/agent/ (id_3)]</refURL> </supervisor> <supervisor> <refURL>[/unifiedconfig/config/agent/ (id_4)]</refURL> </supervisor> </supervisors> </agentTeam> </pre>
Response:	<p>In the Response, the Location header has a URL to the newly created Agent Team, if successful.</p> <p>See also HTTP Responses, on page 7.</p>
Operation Validation:	<p>name</p> <ul style="list-style-type: none"> • Required field. • Max length of 32 bytes allowed. • Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric. • Name must be unique among all Agent Teams in the system. • Does not allow internationalized characters. <p>dialNumber</p> <ul style="list-style-type: none"> • Optional field. • Must refer to a valid Internal Dialed Number. <p>description</p> <ul style="list-style-type: none"> • Optional field. • No restriction of characters. • Max length of 255 bytes allowed.

- For details on valid characters for this field, see [Internationalization](#), on page 9.

agent

- Optional field.
- Must refer to a valid Agent.
- An Agent can be a member of only one Agent Team. **Note:** If an Agent is already on another team, the Agent is removed from that team and put on the team currently being created or updated.
- No more than 50 Agents can be assigned to an Agent Team.

supervisor

- Optional field.
- Must refer to a valid Supervisor.
- A Supervisor can supervise from 0-20 teams.
- A team can have zero to ten Supervisors.
- A Supervisor can also be a member of the Agent Team.

delete

Permanently deletes one Agent Team from the database.

URL:	https://<server>/unifiedconfig/config/agentteam/<id>
HTTP Method:	DELETE
Response:	See .
Operation Validation:	None.

list

Retrieves a list of Agent Teams.

URL:	https://<server>/unifiedconfig/config/agentteam
HTTP Method:	GET

Example XML Response:

```

<results>
  <pageInfo>
    ... (see pagination data)
  </pageInfo>
  <globalInfo>
    <totalVariableSize>1024</totalVariableSize>
  </globalInfo>
  <agentTeams>
    <agentTeam>...</agentTeam>
    <agentTeam>...</agentTeam>
  </agentTeams>
</results>

```

Note The preceding example XML response does not show all of the data for pagination. See [Pagination, on page 10](#).

Note Also, the example XML response does not show permissions information. See [Permissions Information, on page 12](#).

See also [HTTP Responses, on page 7](#).

get

Returns one Agent Team record from the database.

URL:	https://<server>/unifiedconfig/config/agentteam/<id>
HTTP Method:	GET

Example XML Response:

```

<agentTeam>
  <refURL>http://<server>/unifiedconfig/config/agentteam/(id)</refURL>
  <name>team1</name>
  <dialNumber>

<refURL>[https://<server>/unifiedconfig/config/dialednumber/(id)]</refURL>
  <dialNumberString>8885551212</dialNumberString>
</dialNumber>
  <description>test agent team1</description>
  <agents>
    <agent>

<refURL>[https://<server>/unifiedconfig/config/agent/(id_1)]</refURL>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
    <userName>userName</userName>
    <agentId>8006</agentId>
    </agent>
    <agent>

<refURL>[https://<server>/unifiedconfig/config/agent/(id_2)]</refURL>
    <firstName>Jane</firstName>
    <lastName>Doe</lastName>
    <userName>username</userName>
    <agentId>8007</agentId>
    </agent>
  </agents>

  <supervisor>
    <supervisor>

<refURL>[https://<server>/unifiedconfig/config/agent/(id_3)]</refURL>
    <firstName>Mary</firstName>
    <lastName>Hart</lastName>
    <userName>userName</userName>
    <agentId>8008</agentId>
    </supervisor>
  </supervisor>

<refURL>[https://<server>/unifiedconfig/config/agent/(id_4)]</refURL>
    <firstName>Jack</firstName>
    <lastName>Jones</lastName>
    <userName>userName</userName>
    <agentId>8009</agentId>
    </supervisor>
  </supervisors>
  <changeStamp>0</changeStamp>
</agentTeam>

```

See [HTTP Responses](#), on page 7.

update

Updates one Agent Team record in the database.

URL:	https://<server>/unifiedconfig/config/agentteam/<id>
HTTP Method:	PUT
Input/Output Format:	xml

Example XML Request Payload:	<pre> <agentTeam> <name>team1</name> <dialNumber> <refURL>[/unifiedconfig/config/dialednumber/(id)]</refURL> </dialNumber> <description>test agent team1</description> <agents> <agent> <refURL>[/unifiedconfig/config/agent/(id)]</refURL> </agent> </agents> <supervisors> <supervisor> <refURL>[/unifiedconfig/config/agent/(id)]</refURL> </supervisor> </supervisors> <changeStamp>0</changeStamp> </agentTeam> </pre>
Parameters:	See API Parameters , on page 53.
Response:	See HTTP Responses , on page 7.
Operation Validation:	All of the restrictions from the create operation apply. The contents of the <agents> and <supervisors> tags replace the values already configured for the Agent Team.



Attribute API

You can use the Attribute API to list the Attributes currently defined in the database, define new Attributes, and view, edit, or delete records of existing Attributes.

Attributes identify a call routing requirement, such as language, location, or Agent expertise. You can create two types of Attributes: Boolean or proficiency. You can use Boolean Attributes to tag an Attribute that must exist (or must *not* exist). For example, you can create a Boston Attribute that specifies that the Agent assigned to this Attribute must be located in Boston. When you create a proficiency Attribute, you assign a proficiency level to the Attribute.

- [Attribute API Commands](#), page 59

Attribute API Commands

This section explains the five supported API operations for Attribute and outlines the parameters.

API Parameters

The following table shows the parameters for Attribute API:

Parameter Name	Description
appearsOnDesktop	(For future use.)
dataType	Used to assign a data type to the Attribute, using the values: <ul style="list-style-type: none">• 3 = Boolean.• 4 = Proficiency (special form of Integer).
name	Used to assign a unique name for the Attribute.
defaultValue	Used to specify a default value for the Attribute when assigned to an Agent if no explicit value is provided.
settableByAgent	(For future use.)

Parameter Name	Description
description	Used to provide a description for the Attribute.
changeStamp	This parameter represents the current state of the database. This value is populated when the system retrieves data using the GET command.

create

Creates an Attribute and stores the data in the database.

URL:	https://<server>/unifiedconfig/config/attribute
HTTP Method:	POST
Input/Output Format:	xml
Parameters:	See API Parameters , on page 59.
Example XML Request Payload:	<pre><attribute> <dataType>4</dataType> <defaultValue>5</defaultValue> <description>Attribute to specify proficiency in Spanish.</description> <name>Spanish</name> </attribute></pre>
Response:	<p>In the Response, the Location header has a URL to the newly created Attribute, if successful.</p> <p>For example:</p> <p>HTTP/1.1 201 Created</p> <p>Location: https://server/unifiedconfig/config/attribute/5000</p> <p>Content-Type: text/plain</p> <p>Content-Length: 0</p> <p>Date: Tue, 12 Jan 2010 16:00:00 GMT</p> <p>See also HTTP Responses, on page 7.</p>
Operation Validation:	<p>General Restrictions</p> <ul style="list-style-type: none"> You can create a system-wide maximum of 10,000 Attributes. <p>dataType</p> <ul style="list-style-type: none"> Required field. Valid values are: <ul style="list-style-type: none"> 3 = Boolean

	<p>◦ 4 = Proficiency (special form of Integer)</p> <p>name</p> <ul style="list-style-type: none"> • Required field. <p>defaultValue</p> <ul style="list-style-type: none"> • Required field. • For Boolean data types, valid default values are True and False. For proficiency data types, valid default values are 1-10. <p>description</p> <ul style="list-style-type: none"> • Optional field. • For details on valid characters for this field, see Internationalization, on page 9.
--	---

delete

Deletes one Attribute from the database.

URL:	https://<server>/unifiedconfig/config/attribute/5000
HTTP Method:	DELETE
Response:	<p>Example:</p> <p>HTTP/1.1 200 OK</p> <p>Content-Type: text/plain</p> <p>Content-Length: 0</p> <p>Date: Tue, 12 Jan 2010 16:00:00 GMT</p> <p>See also HTTP Responses, on page 7.</p>

list

Retrieves a list of existing Attributes.

URL:	https://<server>/unifiedconfig/config/attribute
HTTP Method:	GET

Example XML Response:	<pre><results> <pageInfo> ... (see pagination data) </pageInfo> <attributes> <attribute>...</attribute> <attribute>...</attribute> </attributes> </results></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10.</p> <p>Note Also, the example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See also HTTP Responses, on page 7.</p>
------------------------------	--

get

Returns one Attribute record from the database.

URL:	https://<server>/unifiedconfig/config/attribute/5000
HTTP Method:	GET
Example XML Response:	<pre><attribute> <dataType>4</dataType> <defaultValue>5</defaultValue> <description>Attribute to specify proficiency in Spanish.</description> <name>Spanish</name> <changeStamp>12</changeStamp> </attribute></pre> <p>See HTTP Responses, on page 7.</p>

update

Updates an existing Attribute in the database.

URL:	https://<server>/unifiedconfig/config/attribute/5000
HTTP Method:	PUT
Input/Output Format:	xml
Parameters:	See API Parameters, on page 59 .
Response:	<p>Example:</p> <pre>HTTP/1.1 200 OK Content-Type: text/plain Content-Length: 0 Date: Tue, 12 Jan 2010 16:00:00 GMT</pre> <p>See HTTP Responses, on page 7.</p>

Operation Validation:

Restrictions:

All of the restrictions from the **create** operation apply, in addition to the following:

- For **dataType**, you cannot modify the data type of an existing Attribute.

changeStamp

- Integer.
- The same value must be returned in an update operation, to ensure the client is working with the latest data set. This value is not required when you create an attribute.



Bucket Intervals

You can use the Bucket Intervals API to add new Bucket Intervals, edit the name of an existing Bucket Interval, get a list of all of the configured Bucket Intervals, and delete existing Bucket Intervals.

This API is represented on the User Interface by the Bucket Intervals Gadget.

- [Bucket Intervals API Commands](#), page 65

Bucket Intervals API Commands

This section explains the five supported API operations for Bucket Intervals and their parameters.

API Parameters

The following table shows the parameters for Bucket Intervals API.

Parameter Name	Description
name	The name of the Bucket Interval. Must be unique. The name is the only field that can be modified.
upperBound1	Required. The first Bucket Interval value.
upperBound2	Optional. The next Bucket Interval value. Must be greater than upperBound1.
upperBound3	Optional. The next Bucket Interval value. Must be greater than upperBound2.
upperBound4	Optional. The next Bucket Interval value. Must be greater than upperBound3.
upperBound5	Optional. The next Bucket Interval value. Must be greater than upperBound4.

Parameter Name	Description
upperBound6	Optional. The next Bucket Interval value. Must be greater than upperBound5.
upperBound7	Optional. The next Bucket Interval value. Must be greater than upperBound6.
upperBound8	Optional. The next Bucket Interval value. Must be greater than upperBound7.
upperBound9	Optional. The next Bucket Interval value. Must be greater than upperBound8.
changeStamp	The change stamp of the Bucket Interval record, which is returned in GET.
override	If this is set to true, then remove Call Type references and mark Bucket Interval as deleted.

create

Creates a Bucket Interval to be stored in the database.

URL:	https://<server>/unifiedconfig/config/bucketinterval
HTTP Method:	POST
Input/Output Format:	XML
Parameters:	See API Parameters , on page 65.
Example XML Response:	<pre><bucketInterval> <name>test</name> <upperBound1>10</upperBound1> <upperBound2>20</upperBound2> <upperBound3>30</upperBound3> <upperBound4>40</upperBound4> <upperBound5>50</upperBound5> <upperBound6>60</upperBound6> <upperBound7>70</upperBound7> <upperBound8>80</upperBound8> <upperBound9>90</upperBound9> </bucketInterval></pre>
Response:	In the response, the Location header has a URL link to the newly created Bucket Interval, if successful. See also HTTP Responses , on page 7.

Operation Validation:	<p>Name</p> <ul style="list-style-type: none"> • Required field. • Max length of 32 bytes allowed. • Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric. • Does not allow internationalized characters. <p>upperBound1</p> <ul style="list-style-type: none"> • Required field. • Must be greater than zero. • Absolute value of integer must range from 0 to 2147483647. <p>upperBound2</p> <ul style="list-style-type: none"> • Optional field. • Must be greater than the previous upperBound field or be left blank. Once one upperBound field is left blank, all remaining upperBound fields must also be blank. • Absolute value of integer must range from 0 to 2147483647. <p>Note The upperBound fields 3-9 have the same restrictions as upperBound2.</p>
------------------------------	---

delete

Deletes one Bucket Interval from the database. If the Bucket Interval is referenced by a Call Type, then an API Warning is returned. You can override this warning by passing an **override=true** query parameter. By default this is set to false.

If **override** is set to **true**, then all Call Type references to the record is unlinked first, then the Bucket Interval is marked as deleted.



Note The delete only *marks* the record for deletion, it does not permanently delete the Bucket Interval.

URL:	https://<server>/unifiedconfig/config/bucketinterval/<id>/
HTTP Method:	DELETE
Parameters:	See API Parameters , on page 65.
Response:	If override=true , see HTTP Responses , on page 7. If override=false or if override is not passed and there are Call Type references: see HTTP Responses , on page 7.

list

Retrieves a list of Bucket Intervals.

URL:	https://<server>/unifiedconfig/config/bucketinterval
HTTP Method:	GET
Example XML Response:	<pre><results> <pageInfo> </pageInfo> <bucketIntervals> <bucketInterval>...</bucketInterval> <bucketInterval>...</bucketInterval> </bucketIntervals> </results></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10.</p> <p>Note Also, the example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See also HTTP Responses, on page 7.</p>

get

Returns one Bucket Interval from the database.

URL:	https://<server>/unifiedconfig/config/bucketinterval/<id>
HTTP Method:	GET
Example XML Response:	<pre><bucketInterval> <refURL>http://***.***.***.***/unified/config/bucketInterval/(id)</refURL> <name>test</name> <upperBound1>10</upperBound1> <upperBound2>20</upperBound2> <upperBound3>30</upperBound3> <upperBound4>40</upperBound4> <upperBound5>50</upperBound5> <upperBound6>60</upperBound6> <upperBound7>70</upperBound7> <upperBound8>80</upperBound8> <upperBound9>90</upperBound9> <changeStamp>0</changeStamp> </bucketInterval></pre> <p>See also HTTP Responses, on page 7.</p>

update

Updates one Bucket Interval in the database.



Note you can only change the Bucket Interval's name.

URL:	https://<server>/unifiedconfig/config/bucketinterval/<id>
HTTP Method:	PUT
Input/Output Format:	xml
Example XML Response:	<pre><bucketInterval> <name>test</name> <changeStamp>0</changeStamp> </bucketInterval></pre>
Parameters:	See API Parameters , on page 65.
Response:	See also HTTP Responses , on page 7.
Operation Validation:	<p>Name</p> <ul style="list-style-type: none"> • Optional field. • Max length of 32 bytes allowed. • Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric. • Does not allow internationalized characters. <p>Note In the Update operation, Name is the only field that you can edit. The upperBound fields 1-9 are not editable.</p> <p>changeStamp</p> <ul style="list-style-type: none"> • Required field.

Asynchronous API

See section on [Asynchronous API](#), on page 20.



Note For Bucket Interval API, the Asynchronous feature is supported only for the create, update, and delete operations.



Bulk Job API

You can use the Bulk Job API to list the Bulk Jobs currently defined in the database, define new Bulk Jobs, and view or delete records of existing Bulk Jobs.

- [Bulk Job API Commands](#), page 71
- [Bulk Operations](#), page 74

Bulk Job API Commands

This section explains the four supported API operations for Bulk Job and outlines the parameters.

API Parameters

The following table shows the parameters for Bulk Job API:

Parameter Name	Description
description	The description of the job specified in the UI.
fileContent	The content of the bulk CSV file.

create

Creates a new Bulk Job record and stores the data in the database.

URL:	https://<server>/unifiedconfig/config/bulkjob
HTTP Method:	POST
Input/Output Format:	xml
Parameters:	See API Parameters , on page 71.

Example XML Request Payload:	<pre><bulkJob> <description>dn create bulk job</description> <fileContent>...</fileContent> </bulkJob></pre>
Response:	<p>In the Response, the Location header has a URL to the newly created Bulk Job, if successful.</p> <p>See also HTTP Responses, on page 7.</p>
Operation Validation:	<p>description</p> <ul style="list-style-type: none"> • Optional field. • Max length of 255 bytes allowed. • No restriction of characters. • For details on valid characters for this field, see Internationalization, on page 9. <p>fileContent</p> <ul style="list-style-type: none"> • Required field. • See also: Bulk Operations, on page 74. <p>The valid input value format is:</p> <pre><fileContent> <![CDATA[put the content of your bulk CSV file here]]> </fileContent></pre> <p>Note Note: Size of Bulk CSV file must be less than or equal to 3 MB.</p>
Bulk Job Execution and Processing:	<p>The following is a description of the execution and processing of a Bulk Job.</p> <ul style="list-style-type: none"> • A new Bulk Job starts out in the queued state and then transitions to the processing state when it starts executing. • If multiple Bulk Jobs are created, only one is executed at a time. Jobs are executed in the order they are created. • The job state is marked as completed successfully if all Bulk operations, such as CREATE or UPDATE, are completed successfully. • If one or more Bulk operation fails, the Bulk Job is marked as failed. • A log file is generated for each Bulk Job. The log contains details of each operation that was executed, as well as a summary at the end of the log file indicating whether the job completed successfully or had failures. • If multiple Bulk Jobs are created, one may be in processing state while others are in queued state. If a system failure occurs during this time, the following failure recovery logic occurs at system startup: jobs in queued state are marked as cancelled, jobs in processing state are marked as failed. This recovery is done before any new job execution occurs.

delete

Deletes one Bulk Job from the database.

URL:	https://<server>/unifiedconfig/config/bulkjob/<id>
HTTP Method:	DELETE
Response:	See HTTP Responses , on page 7.
Operation Validation:	<p>Allows delete only from the Data Server that created it.</p> <p>If a job is finished, or not in processing state, the delete operation deletes the job from the database and deletes the associated CSV and log files.</p> <p>Running jobs cannot be deleted.</p> <p>Note Deleting the Bulk Job does not delete any objects created by it.</p>

list

Retrieves a list of all Bulk Jobs.

URL:	https://<server>/unifiedconfig/config/bulkjob
HTTP Method:	GET

To retrieve a list of all Bulk Jobs for a specific job type:

URL:	https://<server>/unifiedconfig/config/bulkjob?q=jobType:<jobType>
HTTP Method:	GET

Example XML Response:	<pre><results> <pageInfo> ... (see pagination data) </pageInfo> <bulkJobs> <bulkJob>...</bulkJob> <bulkJob>...</bulkJob> </bulkJobs> </results></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10.</p> <p>Note Also, the example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See also HTTP Responses, on page 7.</p>
------------------------------	--

get

Returns the details of a single Bulk Job record from the database.

URL:	https://<server>/unifiedconfig/config/bulkjob/<id>
HTTP Method:	GET
Example XML Response for a Dialed Number Job that is Queued:	<pre><bulkJob> <changeStamp>0</changeStamp> <refURL>/unifiedconfig/config/bulkjob/(id)</refURL> <jobHostName>DS1</jobHostName> <createDateTime>1330441858360</createDateTime> <jobState>1</jobState> <jobType>1</jobType> <description>dn create bulk job</description> </bulkJob></pre> <p>See HTTP Responses, on page 7.</p>
Example XML Response for a Dialed Number create Job, Completed Successfully:	<pre><bulkJob> <changeStamp>0</changeStamp> <refURL>/unifiedconfig/config/bulkjob/(id)</refURL> <jobHostName>DS1</jobHostName> <createDateTime>1330441858360</createDateTime> <startDateTime>1330441858361</createDateTime> <endDateTime>1330441858368</endDateTime> <jobState>3</jobState> <jobType>1</jobType> <description>dn create bulk job</description> <!-- logFile and csvFile only show up if we are on the same DS that created the job and the files actually exist. --> <logFile><refURL>/unifiedconfig/config/bulkjob/(id)/log</refURL></logFile> <csvFile><refURL>/unifiedconfig/config/bulkjob/(id)/csv</refURL></csvFile> </bulkJob></pre>

In the XML responses, as shown in the examples in the preceding table, some of the values returned are ones that are not explicitly set by the Admin. The following provides a description of some of these values:

- **createDateTime:** the time the Bulk Job was submitted, expressed as a value in milliseconds, as returned by the java time command. It indicates the time elapsed from the *zero epoch* value of January 1, 1970, 00:00:00 GMT.
- **jobHostName:** the Windows Computer Name of the Data Server that initiated the Bulk Job.
- **startDateTime:** the time the Bulk Job began executing (transitioned into the processing state).
- **endDateTime:** the time the Bulk Job completed or failed (transitioned out of the processing state).
- **jobState:** the current state of the job, where: 1 = queued, 2 = processing, 3 = succeeded, 4 = failed, 5 = cancelled, 6 = partially succeeded.
- **jobType:** the job type, where: 1 = Dialed Number, 2 = Agent.
- **logFile:** a URL to download the log file for the Bulk Job.
- **csvFile:** a URL to download the CSV file that was originally uploaded in the **fileContent** parameter.

Bulk Operations

The **fileContent** for each bulkJob is populated by importing a .CSV file with specific headers. Each bulkJob type has its own set of headers.

Each API that supports bulkJob provides a template to create a job from. The template can be retrieved online by specifying the job type in the URL:

For example:

`https://<server>/unifiedconfig/config/bulkjob/templates/dialednumber`

`https://<server>/unifiedconfig/config/bulkjob/templates/agent`

Dialed Number Bulk Operation

The headers for the **dialedNumber** bulk operation are the following:

- **operation**
- **dialedNumberString**
- **routingType**
- **description**
- **callTypeName**
- **mediaRoutingDomainName**

The **fileContent** validation is done first by validating that the header is correct. Each header must be present in the exact case and column order as listed. All operations can be done in the same .CSV file.

Descriptions

The following table shows the description for each dialedNumber column name:

Column Name	Description
operation	Available operations are CREATE and UPDATE (case insensitive).
dialedNumberString	The dialedNumberString for this dialedNumber.
routingType	The routingType for this dialedNumber.
description	The description for this dialedNumber.
callTypeName	The name of the callType for this dialedNumber.
mediaRoutingDomainName	The name of the mediaRoutingDomain for this dialedNumber.

CREATE Operation

Operation Validation	operation <ul style="list-style-type: none"> • Required field.
-----------------------------	--

dialedNumberString

- Required field.
- See Dialed Number API, dialedNumberString field, in [create, on page 112](#) section.

routingType

- Required field.
- See Dialed Number API, routingType field, in [create, on page 112](#) section.

description

- Optional field; column must be present, but field can be left blank.
- If the description is left blank, it is set to **BulkJob ID #####**, where ##### represents the ID of the Bulk Job.
- See Dialed Number API, description field, in [create, on page 112](#) section.

callTypeName

- Optional field; column must be present, but field can be left blank.
- If the callTypeName cannot be found, a new Call Type is created.
- The name is set to this field and the description is set to the string **BulkJob ID #####**, where ##### represents the ID of the Bulk Job.
- See Call Type API, name field, in [create, on page 92](#) section.

mediaRoutingDomainName

- Required field if the Routing Type is 4 (Multichannel); otherwise it defaults to Cisco_Voice for Routing types 1-3. See the mediaRoutingDomain field in the Dialed Number API, listed in the [create, on page 112](#) command.
- If specified, must match an existing mediaRoutingDomain name.
- Max length of 32 bytes allowed.
- Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric.
- Internationalized characters not allowed.
- If the name cannot be found, an error is returned.

The following are examples of the .csv file data for the CREATE operation. In each case an explanation of the operation and its associated data is provided.

Example 1

Explanation: the following table shows an example of using the CREATE operation to create a dialedNumber with all fields populated.

dialedNumberString:	123456789
----------------------------	-----------

routingType:	1
description:	New dialedNumber
callTypeName:	CallType1
mediaRoutingDomainName:	CISCO_VOICE

Example 2

Explanation: The following table shows an example of using the CREATE operation to create a dialedNumber without optional callTypeName.

dialedNumberString:	123456788
routingType:	1
description:	New dialedNumber
callTypeName:	(blank)
mediaRoutingDomainName:	CISCO_VOICE

Example 3

Explanation: The following table shows an example of using the CREATE operation to create a dialedNumber without optional description. Note that if callType2 does not exist, it is created.

dialedNumberString:	123456787
routingType:	2
description:	(blank)
callTypeName:	CallType2
mediaRoutingDomainName:	CISCO_VOICE

UPDATE Operation

Operation Validation	<p>operation</p> <ul style="list-style-type: none"> • Required field. <p>dialedNumberString</p> <ul style="list-style-type: none"> • Required field. • Must match an existing dialedNumberString. Note that dialedNumberString is not an editable field. • See Dialed Number API, dialedNumberString field, in create, on page 112 section.
-----------------------------	---

routingType

- Required field.
- Must match existing dailedNumberString/routingType pair. Note that the routingType is not an editable field.
- See Dialed Number API, routingType field, in [create](#), on page 112 section .

description

- Optional field; column must be present.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- If field is equal to ~, stentry is cleared.
- See Dialed Number API, description field, in [create](#), on page 112 section.

callTypeName

- Optional field.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- If field is equal to ~, stentry is cleared.
- If the name cannot be found, a new Call Type is created.
- See Call Type API, name field, in [create](#), on page 92 section.

Note For the UPDATE operation, the tilde (~) works for any optional field.

mediaRoutingDomainName

- Required field if the Routing Type is 4 (Multichannel); otherwise it defaults to Cisco_Voice for Routing types 1-3.
- If specified, must match an existing mediaRoutingDomain name.
- Max length of 32 bytes allowed.
- Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric.
- Internationalized characters not allowed.
- If the name cannot be found, an error is returned.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- If field is equal to ~, stentry is set to the default.

The following are examples of the .csv file data for the UPDATE operation. In each case an explanation of the operation and its associated data is provided.

Example 1

Explanation: The following table shows an example of using the UPDATE operation to update a dialedNumber with all fields populated, and change the callTypeName.

dialedNumberString:	123456789
routingType:	1
description:	New dialedNumber
callTypeName:	CallType2
mediaRoutingDomainName:	CISCO_VOICE

Example 2

Explanation: The following table shows an example of using the UPDATE operation to update dialedNumber, where description ~ means that the previous description is removed.

dialedNumberString:	123456788
routingType:	1
description:	~
callTypeName:	(blank)
mediaRoutingDomainName:	CISCO_VOICE

Example 3

Explanation: The following table shows an example of using the UPDATE operation to update dialedNumber, change description, and specify a blank in callTypeName to indicate that the field should not be changed.

dialedNumberString:	123456787
routingType:	2
description:	Change dialedNumber
callTypeName:	(blank)
mediaRoutingDomainName:	CISCO_VOICE

Agent Bulk Operation

The headers for the **agent** bulk operation are the following:

- **operation**
- **agentId**
- **userName**

- **firstName**
- **lastName**
- **password**
- **loginEnabled**
- **description**
- **agentStateTrace**
- **agentDeskSettingsName**
- **agentTeamName**
- **skillGroups**
- **defaultSkillGroup**
- **attributes**
- **supervisorUserName**
- **domainName**
- **supervisorTeams**

The **fileContent** validation is done first by validating that the header is correct. Each header must be present in the exact case and column order as listed. All operations can be done in the same .CSV file.

For a description of each column, see [Agent API, on page 29](#).

CREATE Operation

Operation Validation:	<p>operation</p> <ul style="list-style-type: none"> • Required field. <p>agentId</p> <ul style="list-style-type: none"> • Optional field; column must be present, but field can be left blank. • Column is auto-generated if left blank. • See Agent API, agentId field, in create, on page 30section. <p>userName</p> <ul style="list-style-type: none"> • Required field. • This is a primary key and cannot be changed after creation. • See Agent API, person.userName field, in create, on page 30section. <p>firstName</p> <ul style="list-style-type: none"> • Required field. • See Agent API, person.firstName field, in create, on page 30section.
------------------------------	---

lastName

- Required field.
- See Agent API, person.lastName field, in [create, on page 30](#) section.

password

- Optional field; column must be present, but field can be left blank.
- See Agent API, person.password field, in [create, on page 30](#) section.

loginEnabled

- Optional field; if not specified, defaults to true.
- See Agent API, person.loginEnabled field, in [create, on page 30](#) section.

description

- Optional field; column must be present, but field can be left blank.
- If the description is left blank, it is set to **BulkJob ID #####**, where ##### represents the ID of the Bulk Job.
- See Agent API, description field, in [create, on page 30](#) section.

agentStateTrace

- Optional field; column must be present, but field can be left blank.
- See Agent API, agentStateTrace field, in [create, on page 30](#) section.

agentDeskSettingsName

- Optional field; column must be present, but field can be left blank.
- If a name is specified and it cannot be found, an error is reported.
- This field is used to find the ID for the Agent Desk Setting to create the ref URL.
- See Agent Desk Settings API, name field, in [create, on page 44](#) section.

agentTeamName

- Optional field; column must be present, but field can be left blank.
- If a name is specified and cannot be found, a new team by the name is created.
- The team's name is set to this field and the description is set to the string **BulkJob ID #####**, where ##### represents the ID of the Bulk Job.
- This field is used to find the ID for the Agent Team Name to create the ref URL.
- See Agent Team API, name field, in [create, on page 53](#) section.

skillGroups

- Optional field; column must be present, but field can be left blank.

- The field is Skill Group name(s), delimited by the semicolon(;) character. For example:
 - sales;support;service
- If a name is specified and cannot be found, a new Skill Group by the name is created.
- The Skill Group's name is set to this field and the description is set to the string **BulkJob ID #####**, where ##### represents the ID of the Bulk Job.
- See Skill Group API, name field, in [create, on page 162](#) section.

defaultSkillGroup

- Optional field; column must be present, but field can be left blank.
- If the field is specified, it must reference a Skill Group defined for the Agent, or an error is reported.
- See Skill Group API, name field, in [create, on page 162](#) section .

attributes

- Optional field; column must be present, but field can be left blank.
- The field is attribute name=value pair(s),delimited by the semicolon (;) character. For example:
 - english=true;sales=7
- The Attribute value can be left blank if the Attribute already exists (the default is used). For example:
 - english;sales
- If a name is specified and cannot be found, an Attribute is created with the name and the value set as the default. (If no value is specified an error is returned.)
- The name is set to this field and the description is set to the string **BulkJob ID #####**, where ##### represents the ID of the Bulk Job.
- Attribute data type is inferred on creation (boolean = 3, integer 1-10 inclusively = 4). All Agent Attributes of the name must match the initial data type.
- See Attribute API, name and value fields, in [create, on page 60](#) section.

supervisorUserName

- Optional field; column must be present, but field can be left blank.
- This field must match an existing Active Directory userName.
- When this field is valid, the Agent's Supervisor flag is set to true.
- See Agent API, name and value fields, in [create, on page 30](#)section.

domainName

- Optional field; column must be present, but field can be left blank.

- This field is the Active Directory Domain name within which the Supervisor userName must exist.
- If empty, the default domain name is used.
- See Agent API, name and value fields, in [create, on page 30](#) section.

supervisorTeams

- Optional field; column must be present, but field can be left blank.
- The field is Agent Team(s), delimited by the semicolon (;) character, that are supervised by this Agent. For example:
 - team1;team2;team3
- If a name is specified and cannot be found, a new team by the name is created.
- The team's name is set to this field and the description is set to the string **BulkJob ID #####**, where ##### represents the ID of the Bulk Job.
- If values are present in this field, and the **supervisorUserName** field is left blank, an error is reported.
- See Agent Team API, name field, in [create, on page 53](#) section.

The following are examples of the .csv file data for the CREATE operation. In each case an explanation of the operation and its associated data is provided.

Example 1

Explanation: The following table shows an example of using the CREATE operation to create an Agent with required fields only.

agentId:	(blank)
userName:	jdoe
firstName:	John
lastName:	Doe
password:	(blank)
loginEnabled:	true
description:	new agent
agentStateTrace:	(blank)
agentDeskSettingsName:	(blank)
agentTeamName:	(blank)
skillGroups:	(blank)

defaultSkillGroup:	(blank)
attributes:	(blank)
supervisorUserName:	(blank)
domainName:	(blank)
supervisorTeams:	(blank)

Example 2

Explanation: The following table shows an example of using the CREATE operation to create an Agent with all fields.

agentId:	123456788
userName:	jadoe
firstName:	Jane
lastName:	Doe
password:	abc123
loginEnabled:	false
description:	new agent
agentStateTrace:	false
agentDeskSettingsName:	ads1
agentTeamName:	teamdoe
skillGroups:	sales;support
defaultSkillGroup:	sales
attributes:	english=true;sales=7
supervisorUserName:	jdoe
domainName:	jdoe.com
supervisorTeams:	teamdoe;teamsmoe

UPDATE Operation

Operation Validation:	<p>operation</p> <ul style="list-style-type: none"> • Required field. <p>agentId</p> <ul style="list-style-type: none"> • Required field. • If field is left blank, the entry is saved with the current value intact; that is, the existing value for that field is kept. If the field is a blank, the userName must reference an existing Agent. • This value must be unique or an error is returned (~ is not allowed). The agentId cannot be changed through bulk update. • See Agent API, agentId, in create, on page 30section. <p>userName</p> <ul style="list-style-type: none"> • Required field. • If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept. If the field is blank, the agentId must reference an existing Agent. • If the agentId is a blank, this field must reference an existing Agent. • If the agentId references an existing Agent, the userName is changed to the contents of this field. • See Agent API, person.userName, in create, on page 30section. <p>firstName</p> <ul style="list-style-type: none"> • Required field. • If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept. • The Agent's first name is changed if the value is not left blank. • See Agent API, person.firstName, in create, on page 30section. <p>lastName</p> <ul style="list-style-type: none"> • Required field. • If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept. • The Agent's last name is changed if the value is not left blank. • See Agent API, person.lastName, in create, on page 30section. <p>password</p>
------------------------------	---

- Optional field; column must be present, but field can be left blank.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- If field is equal to ~, the Agent's password is cleared.
- See Agent API, person.password, in [create, on page 30](#) section.

loginEnabled

- Optional field; if not specified, defaults to true.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- If field is equal to ~, stentry is set to default.
- See Agent API, person.loginEnabled, in [create, on page 30](#) section.

description

- Optional field; column must be present, but field can be left blank.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- If field is equal to ~, stentry is cleared.
- See Agent API, description, in [create, on page 30](#) section.

agentStateTrace

- Optional field; column must be present, but field can be left blank.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- If field is equal to ~, stentry is set to default.
- See Agent API, agentStateTrace, in [create, on page 30](#) section.

agentDeskSettingsName

- Optional field; column must be present, but field can be left blank.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- If field is equal to ~, stentry is cleared.
- If the value changes and the field cannot be found, an error is returned.
- See Agent Desk Settings API, name, in [create, on page 44](#) section.

agentTeamName

- Optional field; column must be present, but field can be left blank.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.

- If field is equal to ~, stentry is cleared.
- If the field changes and the name cannot be found, a new team is created.
- The team's name is set to this field and the description is set to the string **BulkJob ID #####**, where ##### is the ID of the Bulk Job that created the Agent Team.
- See Agent Team API, name, in [create, on page 53](#) section.

skillGroups

- Optional field; column must be present, but field can be left blank.
- The field is Skill Group name(s), delimited by the semicolon (;) character. For example:
 - sales;support;service
- If field is left blank, the stentry is saved with the current value(s) intact; that is, the existing value for that field is kept.
- If field is equal to ~, all Skill Groups is removed from the Agent.
- If a name is specified and cannot be found, a new Skill Group by the name is created.
- The Skill Group's name is set to this field and the description is set to the string **BulkJob ID #####**, where ##### is the ID of the Bulk Job that created the Skill Group.
- See Skill Group API, name, in [create, on page 162](#) section.

defaultSkillGroup

- Optional field; column must be present, but field can be left blank.
- If the field is specified, it must reference a Skill Group defined for the Agent.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept. **Note:** if the value is no longer a Skill Group of the Agent, an error is reported.
- If field is equal to ~, the default Skill Group is removed from the Agent.
- See Skill Group API, name, in [create, on page 162](#) section.

attributes

- Optional field; column must be present, but field can be left blank.
- The field is Attribute name=value pair(s), delimited by the semicolon (;) character. For example:
 - english=true;sales=7
- The Attribute value can be left blank if the Attribute already exists (the default is used). For example:
 - english;sales
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.

- If field is equal to ~, all Attributes are removed from the Agent.
- If a name is specified and cannot be found, an Attribute is created with the name and the value set as the default. (An error is returned if no value is specified.)
- Attribute data type is inferred on creation (boolean = 3, integer 1-10 inclusively = 4). All Agent Attributes of the name must match the initial data type.
- The name is set to this field and the description is set to the string **BulkJob ID #####**, where ##### is the ID of the Bulk Job that created the Attribute.
- See Attribute API, name and value fields, in [create, on page 60](#) section.

supervisorUserName

- Optional field; column must be present, but field can be left blank.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- This field must match an existing Active Directory userName (if it is not ~, or not blank).
- When this field is valid, the Agent's Supervisor flag is set to true.
- If this field is ~, the Agent's Supervisor flag is set to false. **Note:** If any teams are specified in the **supervisorTeams** column, an error is reported.
- See Agent API, name and value fields, in [create, on page 30](#) section.

domainName

- Optional field; column must be present, but field can be left blank.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- This field is the Active Directory Domain name within which the Supervisor userName must exist.
- If this field is ~, it is set to the default domain name (domain of the Unified CCE Data Server).
- See Agent API, name and value fields, in [create, on page 30](#) section.

supervisorTeams

- Optional field; column must be present, but field can be left blank.
- If field is left blank, the stentry is saved with the current value intact; that is, the existing value for that field is kept.
- The field is Agent Team(s), delimited by the semicolon (;) character, that are supervised by this Agent. For example:
 - team1;team2;team3
- If a name is specified and cannot be found, a new team by the name is created.
- The team's name is set to this field and the description is set to the string **BulkJob ID #####**, where ##### is the ID of the Bulk Job that created the Supervisor team.

- If values are present in this field, and the **supervisorUserName** field is set to ~, an error is reported.
- If this field is ~, the Agent is removed as a Supervisor for any teams.
- See Agent Team API, name field, in [create, on page 53](#) section.

The following are examples of the .csv file data for the UPDATE operation. In each case an explanation of the operation and its associated data is provided.

Example 1

Explanation: The following table shows an example of using the UPDATE operation to update an Agent with required fields only.

agentId:	123456789
userName:	(blank)
firstName:	John
lastName:	Doe
password:	(blank)
loginEnabled:	true
description:	updated agent
agentStateTrace:	(blank)
agentDeskSettingsName:	(blank)
agentTeamName:	(blank)
skillGroups:	(blank)
defaultSkillGroup:	(blank)
attributes:	(blank)
supervisorUserName:	(blank)
domainName:	(blank)
supervisorTeams:	(blank)

Example 2

Explanation: The following table shows an example of using the UPDATE operation to update Agent user name, password, and login ability, clear Agent Desk Settings and Agent Team, and save other fields.

agentId:	(blank)
-----------------	---------

userName:	jadoe
firstName:	(blank)
lastName:	(blank)
password:	abc123
loginEnabled:	true
description:	(blank)
agentStateTrace:	(blank)
agentDeskSettingsName:	~
agentTeamName:	~
skillGroups:	(blank)
defaultSkillGroup:	(blank)
attributes:	(blank)
supervisorUserName:	(blank)
domainName:	(blank)
supervisorTeams:	(blank)



Call Type API

You can use the Call Type API to list the Call Types currently defined in the database, define new Call Types, and view, edit, or delete records of existing Call Types.

- [Call Type API Commands, page 91](#)

Call Type API Commands

This section explains the five supported API operations for Call Type and their parameters.

Parameters for Call Type API

The following table shows the parameters for Call Type API:

Parameter Name	Description
name	The name of the Call Type. This name must be unique.
description	Additional information about the Call Type.
serviceLevelThreshold	Maximum time in seconds that a caller should wait before being connected with an Agent.
serviceLevelType	This value indicates how the system calculates the service level. <ul style="list-style-type: none">• NULL = Use the system default.• Ignore Abandoned Calls = 1.• Abandoned Calls have Negative Impact = 2.• Abandoned Calls have Positive Impact = 3.
bucketIntervalID	Identifier of the Bucket Interval level, used for reporting.

Parameter Name	Description
changeStamp	Incremented when the record is changed in the database (read-only).

create

Creates one Call Type and stores it in the database.



Note

Call Type has a capacity rule specified. If you create an object that exceeds the capacity for Call Type, an error will result.

URL:	https://<server>/unifiedconfig/config/calltype/
HTTP Method:	POST
Example XML Request Payload:	<pre><callType> <name>test</name> <description>test call type</description> <serviceLevelThreshold>10</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> <changeStamp>0</changeStamp> <bucketInterval> <refURL>[/unifiedconfig/config/bucketinterval/(id)]</refURL> </bucketInterval> </callType></pre>
Response:	<p>The Location header contains the refURL of the Call Type.</p> <p>Location: https://10.86.135.221/unifiedconfig/config/calltype/<id></p> <p>See also HTTP Responses, on page 7.</p>
Operation Validation:	<p>Restrictions:</p> <p>name</p> <ul style="list-style-type: none"> • Required field. • Max length of 32 bytes allowed. • Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric. • Does not allow internationalized characters. <p>description</p> <ul style="list-style-type: none"> • No restriction of characters. • Max length of 255 bytes allowed. • For details on valid characters for this field, see Internationalization, on page 9.

	<p>serviceLevelThreshold</p> <ul style="list-style-type: none"> • Positive Integers only, but can be NULL (default). <p>serviceLevelType</p> <ul style="list-style-type: none"> • Values can only be NULL (default), 1, 2, 3. <p>bucketIntervalID</p> <ul style="list-style-type: none"> • Positive Integers only, must be one of the defined Bucket Interval IDs, or NULL. <p>changeStamp</p> <ul style="list-style-type: none"> • Integers only: initial value 0. • Not applicable for Create Operations.
--	--

delete

Deletes one Call Type from the database.



Note

The delete only *marks* the record for deletion, it does not permanently delete the Call Type.

URL:	https://<server>/unifiedconfig/config/calltype/<id>
HTTP Method:	DELETE
Response:	See HTTP Responses, on page 7 .
Operation Validation:	<p>Restrictions:</p> <ul style="list-style-type: none"> • You cannot delete any Call Type that is referenced by a script. • You cannot delete any Call Type that is referenced by a Call Type Map. • You cannot delete any Call Type that is referenced by a Dialed Number Map.

list

Retrieves a list of Call Types.

URL:	https://<server>/unifiedconfig/config/calltype
HTTP Method:	GET

Example XML Response:	<pre><results> <pageInfo> ... </pageInfo> <callTypes> <callType>...</callType> <callType>...</callType> </callTypes> </results></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10.</p> <p>Note Also, the example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See also HTTP Responses, on page 7.</p>
------------------------------	--

get

Returns one Call Type from the database.

URL:	https://<server>/unifiedconfig/config/calltype/<id>
HTTP Method:	GET
Example XML Response:	<pre><callType> <refURL>[/unifiedconfig/config/calltype/(id)]</refURL> <name>test</name> <description>test call type</description> <id>5002</id> <serviceLevelThreshold>10</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> <changeStamp>0</changeStamp> <bucketInterval> <refURL>[/unifiedconfig/config/bucketinterval/(id)]</refURL> <name>bucket1</name> </bucketInterval> </callType></pre> <p>See HTTP Responses, on page 7.</p> <p>Note The <id> is the Call Type database ID (callTypeID). It is a read-only field, shown only in the list and get API commands.</p>

update

Updates one Call Type in the database.

URL:	https://<server>/unifiedconfig/config/calltype/<id>
HTTP Method:	PUT

Example XML Request Payload:	<pre><callType> <name>test</name> <description>test call type</description> <serviceLevelThreshold>10</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> <changeStamp>0</changeStamp> <bucketInterval> <refURL>[/unifiedconfig/config/bucketinterval/(id)]</refURL> </bucketInterval> </callType></pre> <p>See HTTP Responses, on page 7.</p>
Operation Validation:	All of the same restrictions of the Create operation apply. See create , on page 92.

Asynchronous API

See section on [Asynchronous API](#), on page 20.



Note

For Call Type API, the Asynchronous feature is supported only for the create, update, and delete operations.



Congestion Control API

You can use the Congestion Control API to list and edit the Congestion Control.

This section explains the supported API operations for Congestion Control and outlines the parameters.



Note

This API is read-only for Packaged CCE deployments.

- [Congestion Control API Parameters, page 97](#)
- [Congestion Control get, page 98](#)
- [Congestion Control update, page 98](#)

Congestion Control API Parameters

The following table shows the parameters for Congestion Control API:

Parameter Name	Description
<code>congestionEnabled</code>	Enable or disable Congestion Control.
<code>congestionTreatmentMode</code>	Mode to handle Congestion. Valid values are integers 1-5: <ul style="list-style-type: none">• 1 = Dialed Number default label is used for call treatment.• 2 = Treat call with Routing client default label.• 3 = Treat call with System default label.• 4 = Terminate with Dialog Fail/RouteEnd.• 5 = Release message to the Routing client.

systemDefaultLabel	Default Label string to treat the calls subjected to Congestion Control.
cpsCapacity	Capacity for Call Per Second.
cpsCapacityDefault	Default Call Per Second Capacity for the current deployment type.

Congestion Control get

Returns the Congestion Control from the database.

Syntax

URL: https://<server>/unifiedconfig/config/congestioncontrol

HTTP Method: GET

Response

See also [General Usage](#).

XML request payload

```
<congestionControl>
  <deploymentType>0</deploymentType>
  <congestionTreatmentMode>1</congestionTreatmentMode>
  <congestionEnabled>true</congestionTreatmentMode>
  <systemDefaultLabel></systemDefaultLabel>
  <cpsCapacity>100</cpsCapacity>
  <cpsCapacityDefault>150</cpsCapacityDefault>
</congestionControl>
```

Congestion Control update

Updates one Congestion Control in the database.

Syntax

URL: https://<server>/unifiedconfig/config/congestioncontrol

HTTP Method: PUT

Input/Output format

xml

Response

See [General Usage](#).

Parameters

See [Congestion Control API Parameters](#), on page 97.

Operation validation

congestionEnabled

- Optional field.
- Has a dependency on **deploymentType**.
- If **deploymentType** is 0, this value has to be 0.
- If **deploymentType** is 7 or 10, this value is 1.
- For other deployment types (not 0, 7, or 10), this field can be 1 or 0.

congestionTreatmentMode

- Optional field.
- Valid values are: 1-5

systemDefaultLabel

- Optional field.
- Max length of 32 characters allowed.
- Internationalized characters not allowed.
- If **congestionTreatmentMode** is set to 3 (Treat call with System default string), then **systemDefaultLabel** must be set.

cpsCapacity

- Optional field.
- Cannot be greater than DEPLOYMENT_MAX_CPS, which is specified in the table below, based on Deployment Type.

cpsCapacityDefault

- Display-only field.
- This value shows the DEPLOYMENT_MAX_CPS, which is specified in the table below, based on Deployment Type.

Deployment Types	Definition	DEPLOYMENT_MAX_CPS
0	General	0
1	NAM	300
2	VRU	300
3	NAM Rogger	150

Deployment Types	Definition	DEPLOYMENT_MAX_CPS
4	ICM Router Logger	115
5	8000 Agents Router Logger	69
6	12000 Agents Router Logger	115
7	Packaged CCE : CCE-PAC-M1	8
8	ICM Rogger	58
9	4000 Agents Rogger	35
10	Packaged CCE: CCE-PAC-M1 Lab Only	1
11	HCS-CC 1000 Agents	8
12	HCS-CC 500 Agents	5
13	UCCE 450 Agents Progger	4



Deployment API

Deployment API is a read-only API. You can use it only to view the current Deployment Type of the installation.

Deployment API does not require a user to authenticate. The API is used to restrict the login options for non-Packaged Contact Center Enterprise deployments.

Deployment API should not be confused with Deployment Type Info API, which is not read-only, and can be used to list and edit the current Deployment Type.

- [Deployment API Commands](#), page 101

Deployment API Commands

get

Returns the Deployment Type from the installation.



Note

This is the only operation supported by Deployment API. It has no parameters.

For a list of valid Deployment Types, see: [Deployment Type Info API](#), on page 103.

URL:	https://<server>/unifiedconfig/config/deployment
HTTP Method:	GET
Example XML Response:	<pre><deployment> <deploymentType>7</deploymentType> </deployment></pre>



Deployment Type Info API

You can use the Deployment Type Info API to list the current system Deployment Type and edit the Deployment Type.

- [Deployment Type Info API Commands, page 103](#)
- [VM Validation, page 108](#)

Deployment Type Info API Commands

This section explains the supported API operations for Deployment Type Info and outlines the parameters.

API Parameters

The following table shows the parameters for Deployment Type API:

Parameter Name	Description	Valid Input Values	Return Value on Get
deploymentType	The type of deployment.	0 (General) 1 (NAM) 2 (VRU) 3 (NAM Rogger) 4 (ICM Router Logger) 5 (8000 Agents Router Logger) 6 (12000 Agents Router Logger) 7 (Packaged CCE: CCE-PAC-M1) 8 (ICM Rogger) 9 (4000 Agents Rogger) 10 (Packaged CCE: CCEPACM1 Lab only) 11 (HCS-CC 1000 Agents) 12 (HCS-CC 500 Agents) 13 (Unified CCE 450 Agents Progger) 14 (HCS-CC 4000 Agents)	0 (General) 1 (NAM) 2 (VRU) 3 (NAM Rogger) 4 (ICM Router Logger) 5 (8000 Agents Router Logger) 6 (12000 Agents Router Logger) 7 (Packaged CCE: CCE-PAC-M1) 8 (ICM Rogger) 9 (4000 Agents Rogger) 10 (Packaged CCE: CCEPACM1 Lab only) 11 (HCS-CC 1000 Agents) 12 (HCS-CC 500 Agents) 13 (Unified CCE 450 Agents Progger) 14 (HCS-CC 4000 Agents)
changeStamp	Incremented when the record is changed in the database.		

get

Returns the current deployment state and the results of the capacity and system validation tests.

URL:	https://<server>/unifiedconfig/config/deploymenttypeinfo
HTTP Method:	GET

Example XML Response:

```
<deploymentTypeInfo>
  <systemValidationStatus>
    <!-- same as serviceability API -->
  </systemValidationStatus>

  <capacityInfo>
    <!-- same as serviceability API -->
  </capacityInfo>

  <!-- Only present if deploymentType is 7 (CCEPACM1) -->
  <vmValidationLogURL>/unifiedconfig/config/deploymenttypeinfo/
    vmvalidation/log</vmValidationLogURL>

  <deploymentType>7</deploymentType>
</deploymentTypeInfo>
```

The example XML response does not show permissions information. See [Permissions Information, on page 12](#).

See [HTTP Responses, on page 7](#).

See also: [System Configuration Validation Output, on page 155](#).

See also: [Capacity, on page 158](#).

To get the VM Validation Log, instead of getting the Deployment Type as detailed in the preceding table, you specify an additional path.

Get VM Validation returns the log file for the last attempt at VM Validation; that is, the log file for attempting to change the Deployment Type to CCEPACM1 (7).

URL:	https://<server>/unifiedconfig/config/deploymenttypeinfo/vmvalidation/log
HTTP Method:	GET
Response:	Returns the contents of the log file, if one exists. If the Deployment Type has not been changed to CCEPACM1, the following message is returned: No VM validation log exists on this AW. To generate one, attempt to configure the system to CCEPACM1.

update

Sets the specified Deployment Type if the system validation check, capacity check, and VM Validation for that Deployment Type pass and are required.

URL:	https://<server>/unifiedconfig/config/deploymenttypeinfo
HTTP Method:	PUT
Input/Output Format:	xml
Example XML Request Payload (All Deployments)	<deploymentTypeInfo> <changeStamp>0</changeStamp> <deploymentType>0</deploymentType> </deploymentTypeInfo>

except Packaged CCE: CCE-PAC-M1):	
Example XML Request Payload (Packaged CCE: CCE-PAC-M1):	<pre><deploymentTypeInfo> <changeStamp>59</changeStamp> <deploymentType>7</deploymentType> <vmHosts> <vmHost> <id>sideA</id> <host>"ESXi Side A hostname or IP"</host> <userName>"username"</userName> <password>"password"</password> </vmHost> <vmHost> <id>sideB</id> <host>"ESXi Side B hostname or IP"</host> <userName>"username"</userName> <password>"password"</password> </vmHost> </vmHosts> </deploymentTypeInfo></pre>
Parameters:	See API Parameters , on page 103.
Response:	See HTTP Responses , on page 7.
Operation Validation:	<p>deploymentType</p> <ul style="list-style-type: none"> • Optional field. • Valid values are: <ul style="list-style-type: none"> ◦ 0 = General ◦ 1 = NAM ◦ 2 = VRU ◦ 3 = NAM Rogger ◦ 4 = ICM Router Logger ◦ 5 = 8000 Agents Router Logger ◦ 6 = 12000 Agents Router Logger ◦ 7 = Packaged CCE : CCE-PAC-M1 ◦ 8 = ICM Rogger ◦ 9 = 4000 Agents Rogger ◦ 10 = Packaged CCE : CCE-PAC-M1 Lab Only ◦ 11 = HCS-CC 1000 Agents ◦ 12 = HCS-CC 500 Agents ◦ 13 = UCCE 450 Agents Progger ◦ 14 = HCS-CC 4000 Agents • If an invalid deploymentType is specified, an API error is thrown.

	<ul style="list-style-type: none"> Switching back to Deployment Type 0 is not allowed because the call processing stops. <p>changeStamp</p> <ul style="list-style-type: none"> Required field. Integers only: start with 0.
<p>Example API Error Response if an Invalid Deployment Type is Specified:</p>	<pre><apiErrors> <apiError> <errorData>deploymentType</errorData> <errorMessage>Invalid deployment type.</errorMessage> <errorType>invalidInput.invalidDeploymentType</errorType> </apiError> </apiErrors></pre>
<p>Example XML Error Response if Capacity Check Fails:</p>	<p>The following example lists a single check that failed. However, if there are multiple capacity check failures, an API error is returned in this list for each of those failures:</p> <pre><apiErrors> <apiError> <errorMessage>You have reached capacity for Reason Code. The limit is 100</errorMessage> <errorType>capacityLimit.maxItems</errorType> </apiError> </apiErrors></pre>
<p>Example XML Error Response if System Validation Check Fails:</p>	<p>Similar to capacity check errors, system validation check errors are returned as individual API errors.</p> <p>The example below shows a multiple rules failure.</p> <pre><apiErrors> <apiError> <errorMessage>System validation check failed for rule TYPE10_NETWORK_VRU_MAP_COUNT. Min = 4, Max = 4, Actual = 0</errorMessage> <errorType>deploymentTypeInfo.invalidSystem</errorType> </apiError> <apiError> <errorMessage>System validation check failed for rule UCM_PIM_COUNT. Min = 1, Max = 1, Actual = 2</errorMessage> <errorType>deploymentTypeInfo.invalidSystem</errorType> </apiError> <apiError> <errorMessage>System validation check failed for rule PG_COUNT. Min = 1, Max = 2, Actual = 4</errorMessage> <errorType>deploymentTypeInfo.invalidSystem</errorType> </apiError> <apiError> <errorMessage>System validation check failed for rule SERVICE_MEMBER_COUNT. Min = 0, Max = 0, Actual = 1</errorMessage> <errorType>deploymentTypeInfo.invalidSystem</errorType> </apiError> </apiErrors></pre>

VM Validation

Overview

VM validation is required for CCE-PAC-M1. When attempting to change the Deployment Type to Packaged CCE: CCE-PAC-M1, the virtual machine infrastructure is validated to ensure the systems and resources are correctly configured. The steps that occur in the validation are as follows:

- 1 User provides hostname or IP, userName, and password for the ESXi servers that are configured for SideA and SideB.
- 2 ESXi server properties are validated for each host.
- 3 A list of the Virtual Machines present on each host are retrieved and matched to the defined profiles. **Note:** since no naming convention is defined, the profile list is literally searched for a match to the VM properties.
- 4 A log file is written to the HD of AW with the contents of the validation attempt.
- 5 If all ESXi properties match, and all required profiles match a Virtual Machine, and no additional Virtual Machines are present, the deployment change is allowed to proceed.
- 6 If any of the preceding conditions fail, an error is reported back to the user.

The validation library is deployment aware and utilizes the vm_validation.xml spring configuration file to define the ESXi server properties and VMs that are validated. The IDs in the file must match those in the vmHost xml; that is, sideA and sideB. See example XML in the preceding [update, on page 105](#) section.

VM Checks

#	Requirement	Type
1	VM - Number of CPUs	Required (Exact Match)
2	CPU Reservation	Required (Exact Match)
3	VM - Exact Memory	Required (Exact Match)
4	VM - Exact Disk Size(s)	Required (Exact Match - Order Independent)
5	VM - Exact Number of Disks	Required (Exact Match)
6	VM - VMWare Tools	Collect and Log Only
7	Host - Vendor	Required (Exact Match)
8	Host - BIOS	Required (Major Version Only; that is: C260.xxxx)
9	Host - ESXi Version	Required (Exact Match)
10	Host - Total Number of VMs	Required (Exact Match)
11	Host - Minimum Number of CPUs	Required

12	Host - Minimum Memory	Required
13	Host - Exact Disk Size	Required (Exact Match - Order Independent)

Errors

The following is a list of errors returned if the host information is incorrect or the VMs are not in a valid layout.

Example API Error Response:

```
-- Error 1: Missing VM Host - Cannot not match profile sideC and cannot find host info for
Side A and B.
<apiErrors>
  <apiError>
    <errorData>deploymentType</errorData>
    <errorMessage>Unable to match host to profile. Host ID: sideC Required Profiles:
[sideA, sideB]</errorMessage>
    <errorType>deploymentTypeInfo.invalidSystem.missingVMHostInfo</errorType>
  </apiError>
</apiErrors>

-- Error 2: Cannot connect to VM Host - Invalid host (ip), userName, or password provided
for side A
<apiErrors>
  <apiError>
    <errorData>deploymentType</errorData>
    <errorDetail xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="vmHostErrorDetail">
      <hosts>
        <hostInfo>
          <host>10.86.141.10</host>
          <id>sideA</id>
          <userName>root2</userName>
        </hostInfo>
      </hosts>
    </errorDetail>
    <errorMessage>Unable to connect to host(s): [[Id: sideA Host:
10.86.141.10]]</errorMessage>
    <errorType>deploymentTypeInfo.invalidSystem.cannotConnectToVMHost</errorType>
  </apiError>
</apiErrors>

-- Error 3: Invalid VM Layout - Side A has valid ESX Host properties, but an invalid layout,
Side B has invalid ESX Host properties, but a valid layout
-- Error 3 generated a log file retrievable at the URL
https://<server>/unifiedconfig/config/deploymenttypeinfo/vmvalidation/log (example below)

<apiErrors>
  <apiError>
    <errorData>deploymentType</errorData>
    <errorDetail xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="invalidVMLayoutErrorDetail">
      <hostPropertiesValid>true</hostPropertiesValid>
      <id>sideA</id>
      <vmLayoutValid>false</vmLayoutValid>
    </errorDetail>
    <errorMessage>The virtual machine host properties or layout is invalid.</errorMessage>
    <errorType>deploymentTypeInfo.invalidSystem.invalidVMLayout</errorType>
  </apiError>
  <apiError>
    <errorData>deploymentType</errorData>
    <errorDetail xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="invalidVMLayoutErrorDetail">
      <hostPropertiesValid>false</hostPropertiesValid>
      <id>sideB</id>
      <vmLayoutValid>true</vmLayoutValid>
    </errorDetail>
  </apiError>
</apiErrors>
```

```
</errorDetail>
<errorMessage>The virtual machine host properties or layout is invalid.</errorMessage>

<errorType>deploymentTypeInfo.invalidSystem.invalidVMLayout</errorType>
</apiError>
</apiErrors>
```

```
Log:
VM Validation Results: Wed May 23 11:25:09 EDT 2012
Overall: true
Valid Systems: 2 of 2
Summary:
ESX Server: sideA
ESX Server Properties Valid: true
VM Layout Valid: true
Server Result:
Required Version: 5.0.0
Required Min CPU Cores: 20
Required Min Memory (MB): 95000
Required HD(s) (GB): [1392, 1949, 273]
Required Bios <Major version>: C260
Required Vendor: Cisco Systems Inc
Found Version: 5.0.0
Found CPU Cores: 20
Found Memory (MB): 98185
Found HD(s) (GB): [1392, 273, 1949]
Found Bios: C260.1.4.2b.0.102620111637
Found Vendor: Cisco Systems Inc
```

```
Virtual Machines Matching Defined Profiles:
VM: BB-CCE-DataSvr-A
Profile: Unified CCE Data Server
OS: Microsoft Windows Server 2008 R2 (64-bit)
CPU Cores: 4
Reservation: 5100
RAM (MB): 8192
HD(s) (GB): [80, 750, 500]
VMWare Tools Version: 8384
```

```
Required Profiles without Matching Virtual Machines: None
Optional Profiles without Matching Virtual Machines: None
Virtual Machines without Matching Profiles: None
```




Dialed Number API

You can use the Dialed Number API to list the dialed numbers currently defined in the database, define new dialed numbers, and view, edit, or delete records of existing dialed numbers.



Note

Multiple dialed number records are created for each *external* dialed number, with one for each routing client.

- [Dialed Number API Commands, page 111](#)

Dialed Number API Commands

This section explains the five supported API operations for Dialed Number and their parameters.

API Parameters

The following table shows the parameters for Dialed Number API:

Parameter Name	Description
<code>dialedNumberString</code>	The Dialed Number string.
<code>description</code>	Additional information about the Dialed Number.

Parameter Name	Description
routingType	<p>1 (External Voice), 2 (Internal Voice), 3 (Outbound), 4 (Multichannel)</p> <p>Note Externally Dialed Numbers are Dialed Numbers that apply to calls coming from Unified CVP, which create four separate Dialed Number Records, one for each of the Unified CVP Routing Clients. The calls are referred to as external because typically they come from outside of the enterprise through a gateway.</p> <p>Note Internally Dialed Numbers are Dialed Numbers that can be called from a Unified CM phone. The calls must have a Route Point on Unified CM that corresponds to the Internally Dialed Number. They are referred to as internal because they can only be accessed by Unified CM. Each Internally Dialed Number creates a single Dialed Number Record tied to the Unified CM routing client.</p> <p>Outbound Dialed Numbers are used by the Outbound Option Dialer. They are referenced when creating the outbound campaign configuration in the Campaign Skill Group selection.</p> <p>Multichannel Dialed Numbers are used by Email and Web Collaboration.</p> <p>A Dialed Number Record is a record stored in the configuration database.</p>
changeStamp	Incremented when the record is changed in the database (read-only).
callType refURL	A reference to a Call Type.
Dialed Number Record(s)	Routing client names that the Dialed Number string is associated with (read-only).
mediaRoutingDomain	Media Routing Domain for the Dialed Number.

create

Creates one Dialed Number and stores it in the database.



Note

Creation of an *external* Dialed Number will use every routing client in the system to create that number of routing clients, even if there are not four routing clients. If there are not four, this fact is logged.



Note If the `<dialedNumberRecord>` element is supplied, it is ignored, because it is read-only, and only available on GET.



Note Dialed Number has a capacity rule specified. If you create an object that exceeds the capacity for Dialed Number, an error will result.

URL:	https://<server>/unifiedconfig/config/dialednumber/
HTTP Method:	POST
Parameters:	See API Parameters , on page 111.
Example XML Request Payload:	<pre><dialedNumber> <description>test dialed number</description> <dialedNumberString>8885551212</dialedNumberString> <routingType>1 2</routingType> <callType> <refURL>[/unifiedconfig/config/calltype/(id)]</refURL> </callType> </dialedNumber></pre>
Response:	<p>In the Response, the Location header has a URL to the newly created Dialed Number, if successful.</p> <pre>HTTP/1.1 201 Created Content-Length: 0 Content-Type: text/plain Location: https://<server>/unifiedconfig/config/dialedNumber/(id) Server: Microsoft-IIS/7.5 X-Powered-By: ASP.NET Date: Tue, 18 Oct 2011 10:31:31 GMT</pre> <p>See also HTTP Responses, on page 7.</p>
Operation Validation:	<p>dialedNumberString</p> <ul style="list-style-type: none"> • Required field. • Valid values: alphanumeric, +, and @. • Max length of 25 bytes allowed (Seven reserved internally for EnterpriseName prefix that identifies the routing client). <p>Note Note: The dialedNumberString could also be <i>OB</i>, to refer to an outbound Skill Group, for example.</p> <p>description</p> <ul style="list-style-type: none"> • Optional field. • Max length of 255 bytes allowed. • No restriction of characters. • For details on valid characters for this field, see Internationalization, on page 9.

routingType

- Required field.
- Values are:
 - 1 (External Voice)
 - 2 (Internal Voice)
 - 3 (Outbound)
 - 4 (Multichannel)

callType

- Optional field.
- Must be a refURL for a valid Call Type.

mediaRoutingDomain

- Optional field for Routing Types 1-3; if specified, must be Cisco_Voice.
- Required field for Routing Type 4 (Multichannel); if specified, can be Cisco_Voice or a user-defined value.

Note The user-defined type must be defined in the Media Routing Domain table.

delete

Deletes one Dialed Number from the database.



Note The delete command not only deletes **t_Dialed_Number** records, but also any **t_Dialed_Number_Map** records related to the **t_Dialed_Number**.



Note The delete only *marks* the record for deletion, it does not permanently delete the Dialed Number.

URL:	https://<server>/unifiedconfig/config/dialednumber/<id>
HTTP Method:	DELETE
Response:	See also HTTP Responses , on page 7.
Operation Validation:	You cannot delete any Dialed Number that is: <ul style="list-style-type: none"> • Referenced by an Agent Team. • Referenced by an Agent's Desk Settings. • Referenced by a script.

list

Retrieves a list of Dialed Numbers.

**Note**

For *external* Dialed Numbers, the **list** command only lists the Dialed Number with the *lowestDialedNumberID* value.

URL:	https://<server>/unifiedconfig/config/dialednumber
HTTP Method:	GET
Example XML Response:	<pre><results> <pageInfo> ... </pageInfo> <dialedNumbers> <dialedNumber>...</dialedNumber> <dialedNumber>...</dialedNumber> </dialedNumbers > </results></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10.</p> <p>Note Also, the example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See also HTTP Responses, on page 7.</p>
routingType:<type> Search Parameter:	<p>In addition to the standard search, q=string, which searches the dialedNumberString and description fields, the Dialed Number API also supports adding the q=routingType:<type> string to the query string.</p> <p>The following restrictions apply:</p> <ul style="list-style-type: none"> Valid types are the integers: <ul style="list-style-type: none"> ◦ 1 (External Voice) ◦ 2 (Internal Voice) ◦ 3 (Outbound) ◦ 4 (Multichannel) Only one type can be specified in a query. You can add the routingType:<type> query to a regular search query. For example: q=routingType:1 123. The routingType:<type> parameter must be the first parameter of the search string.

get

Returns one Dialed Number from the database.

**Note**

In the case of an *external* Dialed Number, you must specify the smallest Dialed Number for that bank of Dialed Numbers to retrieve it.

URL:	https://<server>/unifiedconfig/config/dialednumber/<id>
HTTP Method:	GET
Example XML Response:	<pre><dialedNumber> <refURL>[/unifiedconfig/config/dialedNumber/(id)]</refURL> <description>test dialed number</description> <dialedNumberString>8885551212</dialedNumberString> <routingType>1</routingType> <changeStamp>0</changeStamp> <mediaRoutingDomain> <refURL>[/unifiedconfig/config/mediaroutingdomain/1]</refURL> <name>Cisco_Voice</name> </mediaRoutingDomain> <callType> <refURL>[/unifiedconfig/config/calltype/(id)]</refURL> <name>calltype name</name> </callType> <dialedNumberRecords> <dialedNumberRecord> <id>10</id> <name>cvp1rc.8885551212</name> </dialedNumberRecord> <dialedNumberRecord> <id>11</id> <name>cvp2rc.8885551212</name> </dialedNumberRecord> <dialedNumberRecord> <id>12</id> <name>cvp3rc.8885551212</name> </dialedNumberRecord> <dialedNumberRecord> <id>13</id> <name>cvp4rc.8885551212</name> </dialedNumberRecord> </dialedNumberRecords> </dialedNumber></pre> <p>See HTTP Responses, on page 7.</p>
Exceptions:	Note If retrieving an <i>external</i> (IVR) Dialed Number and not all of the four t_Dialed_Number records are present, an error is logged to the Tomcat log, and an HTTP status code of 404 (not found) is returned to the caller.

update

Updates one Dialed Number in the database.

**Note**

If the **<dialedNumberRecord>** element is supplied, it is ignored, as it is read-only, and only available on GET.

URL:	https://<server>/unifiedconfig/config/dialednumber/<id>
HTTP Method:	PUT

Example XML Request Payload:	<pre><dialedNumber> <description>test dialed number</description> <dialedNumberString>8885551212</dialedNumberString> <routingType>1 2</routingType> <changeStamp>0</changeStamp> <callType> <refURL>[/unifiedconfig/config/calltype/(id)]</refURL> <name>calltype name</name> </callType> </dialedNumber></pre>
Response:	See HTTP Responses , on page 7.
Operation Validation:	<p>All of the restrictions from the preceding create, on page 112 section apply, in addition to the following restrictions:</p> <p>refURL</p> <ul style="list-style-type: none"> • Required field. • Must refer to a valid Dialed Number. <p>changeStamp</p> <ul style="list-style-type: none"> • Required field. • Integers only: start with 0.

Asynchronous API

See section on [Asynchronous API](#), on page 20.



Note

For Dialed Number API, the Asynchronous feature is supported only for the create, update, and delete operations.



Expanded Call Variable APIs

You can use the Expanded Call Variable API to list the Expanded Call Variables currently defined in the database, define new Expanded Call Variables, and view, edit, or delete records of existing Expanded Call Variables.



Note

Expanded Call Variables are also called Expanded Call Context (ECC) Variables.

- [Expanded Call Variable API Commands, page 119](#)

Expanded Call Variable API Commands

This section explains the five supported API operations for Expanded Call Variable and their parameters.

API Parameters

The following table shows the parameters for Expanded Call Variable API:

Parameter Name	Description
name	The name of the Expanded Call Variable. This name must be unique.
maxLength	The maximum length of the Expanded Call Variable value: 1 to 210.
eccArray	True/False. Indicates whether the Expanded Call Variable is an array.
maximumArraySize	If the Expanded Call Variable is an array, the maximum number of elements in the array: 1 to 255.
enabled	True/False. Indicates whether the Expanded Call Variable is currently enabled.

Parameter Name	Description
description	Additional information about the Expanded Call Variable.
persistent	True/False. Specifies whether each individual Expanded Call Variable is persistent (is written to the historical database with the TCD or RCD record).
ciscoProvided	True/False. Indicates whether the Expanded Call Variable is provided by Cisco.
changeStamp	Incremented when the record is changed in the database.
bytesRequired	Indicates the number of bytes required to store the Expanded Call Variable in the system. This field is only valid for get and list operations.
bytesRequiredInCtiServer	The same parameters as bytesRequired , but applies to CTI Server. In CTI Server, the number of bytes required includes the length of the Expanded Call Variable name.

create

Creates one Expanded Call Variable and stores it in the database.

URL:	https://<server>/unifiedconfig/config/expandedcallvariable
HTTP Method:	POST
Input/Output Format:	xml
Parameters:	See .
Example XML Request Payload:	<pre><expandedCallVariable> <name>test</name> <maxLength>210</maxLength> <maximumArraySize>255</maximumArraySize> <eccArray>true</eccArray> <enabled>true</enabled> <description>test expanded call variable</description> <persistent>false</persistent> <changeStamp>0</changeStamp> </expandedCallVariable></pre>
Response:	In the Response, the Location header has a URL to the newly created Expanded Call Variable, if successful. See also HTTP Responses , on page 7.
Operation Validation:	General

- The maximum number of persistent, scalar, enabled variables is 20.
- No persistent, enabled arrays are allowed.
- The size of an Expanded Call Variable cannot exceed 2000 bytes, regardless of whether it is enabled.
- The total size of all enabled Expanded Call Variables cannot exceed 2000 bytes.
- The size of an Expanded Call Variable is calculated using the following formula:
 - If eccArray is false, the size is 5+Maximum Length.
 - If eccArray is true, the size is 5+(1+Maximum length)*Maximum Array size.
- The size of an Expanded Call Variable in CTI Server cannot exceed 2500 bytes, regardless of whether it is enabled.
- The total size of all enabled Expanded Call Variables in CTI Server cannot exceed 2500 bytes.
- The size of an Expanded Call Variable in CTI Server is calculated using the following formula:
 - If eccArray is false, the size is Length of name+Maximum length+4.
 - If eccArray is true, the size is (Length of name+Maximum length+5)*Maximum array size.

name

- Required field.
- Max length of 32 bytes allowed.
- Must begin with "user".
- Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric.
- Name must be unique.
- Does not allow internationalized characters.

maximumLength

- Required field.
- Value must be between 1 and 210.

maximumArraySize

- Value must be between 1 and 255.
- Required if eccArray is true.
- Must be blank or not specified if eccArray is false.

eccArray

- Boolean: true/false (default is false).

enabled

- Boolean: true/false (default is true).

ciscoProvided

- Boolean: true/false (read-only).
- If true, everything but Enabled and Persistent are read-only.

description

- No restriction of characters.
- Max length of 255 bytes allowed.
- For details on valid characters for this field, see [Internationalization, on page 9](#).

persistent

- Boolean: true/false (default is false).

changeStamp

- Integers only: initial value 0.
- Not applicable for Create Operations.

delete

Deletes one Expanded Call Variable from the database.

**Note**

The delete only *marks* the record for deletion, it does not permanently delete the Expanded Call Variable.

URL:	https://<server>/unifiedconfig/config/expandedcallvariable/<id>/
HTTP Method:	DELETE
Response:	See also HTTP Responses, on page 7 .
Operation Validation:	You cannot delete any Expanded Call Variable that is: <ul style="list-style-type: none"> • Cisco Provided. • Referenced by a script.

list

Retrieves a list of Expanded Call Variables.

URL:	https://<server>/unifiedconfig/config/expandedcallvariable
HTTP Method:	GET
Example XML Response:	<pre><results> <pageInfo> ... </pageInfo> <globalInfo> <totalVariableSize>1024</totalVariableSize> </globalInfo> <expandedCallVariables> <expandedCallVariable>...</expandedCallVariable> <expandedCallVariable>...</expandedCallVariable> </expandedCallVariables> </results></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10.</p> <p>Note Also, the example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See also HTTP Responses, on page 7.</p>

get

Returns one Expanded Call Variable from the database.

URL:	https://<server>/unifiedconfig/config/expandedcallvariable/<id>
HTTP Method:	GET
Example XML Response:	<pre><expandedCallVariable> <refURL>/unifiedconfig/config/expandedcallvariable/(id)</refURL> <name>test</name> <maxLength>9</maxLength> <maximumArraySize>10</maximumArraySize> <eccArray>true</eccArray> <enabled>true</enabled> <ciscoProvided>>false</ciscoProvided> <description>test expanded call variable</description> <persistent>>false</persistent> <changeStamp>0</changeStamp> <bytesRequired>105</bytesRequired> <bytesRequiredInCtiServer>180</bytesRequiredInCtiServer> </expandedCallVariable></pre> <p>See HTTP Responses, on page 7.</p>

update

Updates one Expanded Call Variable in the database.

URL:	https://<server>/unifiedconfig/config/expandedcallvariable/<id>
-------------	---

HTTP Method:	PUT
Input/Output Format:	xml
Example XML Request Payload:	<pre><expandedCallVariable> <name>test</name> <maxLength>210</maxLength> <enabled>true</enabled> <description>test expanded call variable</description> <persistent>false</persistent> <changeStamp>0</changeStamp> </expandedCallVariable></pre>
Parameters:	See API Parameters , on page 119.
Response:	See HTTP Responses , on page 7.
Operation Validation:	<p>All of the restrictions from the Create operation apply, in addition to the following restrictions:</p> <p>MaximumArraySize</p> <ul style="list-style-type: none"> • Cannot be updated. <p>eccArray</p> <ul style="list-style-type: none"> • Cannot be updated. <p>enabled</p> <ul style="list-style-type: none"> • Cannot be set to false if the Expanded Call Variable is referenced by a script.

Asynchronous API

See section on [Asynchronous API](#), on page 20.



Note

For Expanded Call Variable API, the Asynchronous feature is supported only for the create, update, and delete operations.



Media Routing Domain API

A Media Routing Domain is a collection of Skill Groups and services associated with a common media class. It is used to organize how requests for different media are routed.

You can use the Media Routing Domain (MRD) API to list the MRDs currently defined in the database and view records of MRDs.

Note: The operations of create, delete, and update are not supported for the Media Routing Domain API.

- [Media Routing Domain API Commands](#), page 125

Media Routing Domain API Commands

This section describes the two supported API operations for Media Routing Domain.

list

Retrieves a list of Media Routing Domains.

URL:	https://<server>/unifiedconfig/config/mediaroutingdomain
HTTP Method:	GET
Example XML Response:	<pre><results> <pageInfo> ... (see pagination data) </pageInfo> <mediaRoutingDomains> <mediaRoutingDomain>...</mediaRoutingDomain> <mediaRoutingDomain>...</mediaRoutingDomain> </mediaRoutingDomains> </results></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10. See also HTTP Responses, on page 7.</p>
nonVoiceOnly Search Parameter:	Set the nonVoiceOnly Attribute to true in the search query parameter to make the API return only Media Routing Domains other than the Cisco_Voice Media Routing Domain.

For example: **q=nonVoiceOnly:true**

get

Returns one Media Routing Domain record from the database.

URL:	https://<server>/unifiedconfig/config/mediaroutingdomain/<id>
HTTP Method:	GET
Example XML Response:	<pre><mediaRoutingDomain> <refURL>/unifiedconfig/config/mediaroutingdomain/(id)</refURL> <name>Cisco_Voice_MRD</name> <description>this is a mrd</description> <interruptible>false</interruptible> <mediaClass> <name>Cisco_Voice</name> <id>4</id> </mediaClass> <serviceLevelThreshold>30</serviceLevelThreshold> </mediaRoutingDomain></pre> <p>See HTTP Responses, on page 7.</p>
Parameters:	<p>Media Class: Name and ID</p> <p>Service Level Threshold: Integer. Value in seconds within which calls are to be answered.</p> <p>Interruptible: True False</p>



Network VRU Script APIs

You can use the Network VRU Script API to list, create, edit and delete Network VRU Scripts.

- [Network VRU Script API Commands, page 127](#)

Network VRU Script API Commands

This section explains the five supported API operations for Network VRU Script and their parameters.

API Parameters

The following table shows the parameters for Network VRU Script API:

Parameter Name	Description
routingType	1 (Voice), 2 (Multichannel). This field is optional and defaults to 1. Voice network VRU scripts are used by Unified CVP. Multichannel network VRU scripts are used by Email and Web Collaboration.
name	The name of the Network VRU as seen by CCE. This name must be unique.
vruScriptName	The name of the network VRU script as seen by the network VRU.
timeout	Number of seconds for the system to wait for a response from the routing client after directing it to run the script.
configParam	Optional string sent to the VRU to initialize the script.
interruptible	Indicates whether the script can be interrupted. Y/N.
description	Additional information about the script.

Parameter Name	Description
changeStamp	Incremented when the record is changed in the database (read-only).

create

Creates one network VRU script and stores it in the database.

URL:	https://<server>/unifiedconfig/config/networkvruscript
HTTP Method:	POST
Input/Output Format:	xml
Parameters:	See API Parameters , on page 127.

Example XML Response:	<pre><networkVruScript> <routingType>1</routingType> <name>test</name> <vruScriptName>GS, Server, V</vruScriptName> <timeout>180</timeout> <configParam>Y</configParam> <interruptible>Y</vruScriptName> <description>CVP VXML Server script</description> <changeStamp>0</changeStamp> </networkVruScript></pre> <p>In the response, the Location header has a URL to the newly created network VRU script, if successful.</p>
Response:	<pre>HTTP/1.1 201 Created Content-Length: 0 Content-Type: text/plain Location: https://<server>/unifiedconfig/config/networkvruscript/(id) Server: Microsoft-IIS/7.5 X-Powered-By: ASP.NET Date: Tue, 18 Oct 2011 10:31:31 GMT</pre> <p>See also HTTP Responses, on page 7.</p>
Operation Validation:	

name

- Required field.
- Max length of 32 bytes allowed.
- Valid values: alphanumeric, +, and @.

vruscriptName

- Required field.
- Max length of 39 bytes allowed.
- Valid values: free form.
- Must be unique for a given routing type.

timeout

- Required field.
- Valid values: 1- max integer, default 180.
- Max length: max integer.

configParam

- Optional field.
- Valid values: free form.
- Max length: 255.

interruptible

- Optional field.
- Valid values: true/false.

description

- Optional field.
- Valid values: free form.
- Max length: 255.
- For details on valid characters for this field, see [Internationalization](#), on page 9.

routingType

- Optional field.
- Valid values: 1 or 2.

delete

Deletes one Network VRU Script from the database.

**Note**

Network VRU Scripts may be referenced in routing scripts. If this is the case, an error is returned.

URL:	https://<server>/unifiedconfig/config/networkvru-script/<id>
HTTP Method:	DELETE
Response:	See HTTP Responses , on page 7.

list

Retrieves a list of Network VRU Scripts.

URL:	https://<server>/unifiedconfig/config/networkvru-script
HTTP Method:	GET
Example XML Response:	<pre><paginationInfo> <resultsPerPage>2</resultsPerPage> <startIndex>0</startIndex> <totalResults>10</totalResults> <firstPage> http://<server>/networkvru-script/?resultsPerPage=2</firstPage> <lastPage> http://<server>/networkvru-script/?startIndex=8&resultsPerPage=2</lastPage> <prevPage/> <nextPage> http://<server>/networkvru-script/?startIndex=2&resultsPerPage=2</nextPage> </paginationInfo><networkVruScripts> <networkVruScript>...</networkVruScript> <networkVruScript>...</networkVruScript> </networkVruScripts></pre> <p>Note The preceding example XML response does not show all of the data for pagination. See Pagination, on page 10.</p> <p>Note Also, the example XML response does not show permissions information. See Permissions Information, on page 12.</p> <p>See also HTTP Responses, on page 7.</p>

get

Returns one Network VRU Script from the database.

URL:	https://<server>/unifiedconfig/config/networkvru-script/<id>
HTTP Method:	GET

Example XML Response:	<pre> <networkVruScript> <refURL>http://***.***.***.***/unified/config/networkvruScript/(id)</refURL> <routingType>1</routingType> <name>test</name> <vruScriptName>GS, Server, V</vruScriptName> <timeout>180</timeout> <configParam>Y</configParam> <interruptible>true</interruptible> <description>CVP VXML Server script</description> <changeStamp>0</changeStamp> </networkVruScript> </pre> <p>See HTTP Responses, on page 7.</p>
------------------------------	---

update

Updates one network VRU script in the database.

URL:	https://<server>/unifiedconfig/config/networkvruScript/<id>
HTTP Method:	PUT
Example XML Response:	<pre> <networkVruScript> <routingType>1</routingType> <name>test</name> <vruScriptName>GS, Server, V</vruScriptName> <timeout>180</timeout> <configParam>Y</configParam> <interruptible>true</interruptible> <description>CVP VXML Server script</description> <changeStamp>0</changeStamp> </networkVruScript> </pre>

Asynchronous API

See section on [Asynchronous API](#), on page 20.



Note

For Network VRU Script API, the Asynchronous feature is supported only for the create, update, and delete operations.



Precision Queue API

You can use the precision queue API to list the precision queues currently defined in the database, define new precision queues, and view, edit, or delete records of existing precision queues.

Precision queues are a combination of steps that include Attributes, defined terms for the selected Attributes, and wait times. A precision queue step is a time-based routing point within the precision queue.

You must have at least one step and can have up to ten steps. Within each step you can have:

- Up to five unique Attributes
- A different pool of Agents
- A wait time

A precision queue term compares an Attribute against a value. For example, if you have an Attribute for English (English_skill) and assign it a value of > 7 , the term is `English_skill > 7`. You can create a maximum of 2000 system-wide precision queues. You can associate a maximum of ten precision queue terms to a precision queue step. You can associate a maximum of five Attributes to a precision queue term.

The following limits are enforced through the API:

- You can add a maximum of 2000 precision queues to the entire system.
- You can associate a maximum of ten precision queue terms to a precision queue step.
- You can associate up to five unique Attributes to a precision queue.
- An Agent can be a member of a combination of up to 50 precision queues and Skill Groups.
- [Precision Queue API Commands, page 133](#)

Precision Queue API Commands

This section explains the supported API operations for precision queue and outlines the parameters.

API Parameters

The following table shows the precision queue parameters:

Parameter Name	Description
bucketInterval	A reference to a bucketInterval. See the bucketInterval parameters below.
agentCount	An optional parameter that returns Agent Count for the precision queue and each step.
agentOrdering	Determines the order of Agents in a precision queue sub-queue using the following values: <ul style="list-style-type: none"> • 1 = LAA (Agent availability time). • 2 = Most skilled Agent. • 3 = Least skilled Agent.
callOrdering	Determines the order of calls in this precision queue using the following value: <ul style="list-style-type: none"> • 1 = Priority, then time in queue.
changeStamp	This parameter represents the current state of the database. The value is populated when the system retrieves data using the GET command.
name	A unique enterprise name for the precision queue.
description	Description for the precision queue.
serviceLevelThreshold	The service level threshold in seconds for this precision queue.
serviceLevelType	Determines how to calculate the service level for the precision queue using the following values: <ul style="list-style-type: none"> • 1 = Ignore abandoned calls. • 2 = Abandoned call has negative impact. • 3 = Abandoned call has positive impact.
steps	A list of steps for this precision queue. See the Step Parameters.

The following table shows the bucketInterval parameters:

Parameter Name	Description
name	The unique enterprise name of the Bucket Interval.
refURL	The URL of the referenced Bucket Interval.

You can create up to ten steps for each precision queue. The following table shows the step parameters:

Parameter Name	Description
stepOrder	The order of rows for a precision queue step.
waitTime	A wait time to apply before proceeding to the next step (in seconds).
considerIf	Consider If expression which must be met to execute a particular step.
description	Description for this step.
terms	A list of terms for this step. See the term parameters.

You can configure up to ten terms for each step. The following table shows the term parameters:

Parameter Name	Description
termOrder	The order of rows for a precision queue term.
attribute	A reference to the Attribute to be tested.
parenCount	The number of parenthesis around this term. A positive number indicates open parens before the term; a negative number indicates closed parens after the term.
termRelation	Indicates the relationship of this term to the preceding term, using the following values: <ul style="list-style-type: none"> • 0 = None (only legal on first item for queue, since there is no preceding term). • 1 = AND. • 2 = OR.
attributeRelation	Indicates what kind of comparison is done on the Attribute, using the following values: <ul style="list-style-type: none"> • 1 = Equal. • 2 = Not equal. • 3 = Less than. • 4 = Less than or equal. • 5 = Greater than. • 6 = Greater than or equal.

Parameter Name	Description
value1	The value that the Attribute is tested against. It must be convertible to the data type specified in the Attribute table.

create

Creates a precision queue, precision queue steps, and precision queue terms.

URL:	https://<server>:port/unifiedconfig/config/precisionqueue
HTTP Method:	POST
Input/Output Format:	xml
Parameters:	See API Parameters, on page 133 .
Example XML	<pre><precisionQueue> <bucketInterval> <markDeletable>true</markDeletable> </bucketInterval> </precisionQueue></pre>

Request Payload:	<pre> <name>huiTest</name> <refURL>[http://bos-cl-cc:8080/unifiedconfig/config/bucketinterval/5000]</refURL> </bucketInterval> <agentOrdering>1</agentOrdering> <callOrdering>1</callOrdering> <description>sales precision queue</description> <name>SalesAQ</name> <scriptModificationAllowed>>false</scriptModificationAllowed> <serviceLevelThreshold>1</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> <steps> <step> <stepOrder>1</stepOrder> <waitTime>-1</waitTime> <considerIf>test</considerIf> <nextStep>>false</nextStep> <description>"Step 1"</description> <terms> <term> <attribute> <refURL>http://bos-cl-cc:8080/unifiedconfig/config/attribute/5033</refURL> <name>Language</name> <dataType>4</dataType> <description>Marketing</description> </attribute> <termOrder>1</termOrder> <parenCount>1</parenCount> <termRelation>0</termRelation> <attributeRelation>5</attributeRelation> <value1>2</value1> <value2>10</value2> </term> <term> <attribute> <refURL>http://bos-cl-cc:8080/unifiedconfig/config/attribute/5033</refURL> <name>Language</name> <dataType>4</dataType> <description>Marketing</description> </attribute> <termOrder>2</termOrder> <parenCount>-1</parenCount> <termRelation>1</termRelation> <attributeRelation>5</attributeRelation> <value1>5</value1> <value2>6</value2> </term> </terms> </step> </steps> </precisionQueue> </pre>
HTTP Response Headers:	<p>In the Response, the Location header has a URL to the newly created precision queue, if successful.</p> <p>For example:</p> <pre> HTTP/1.1 201 Created http://bos-cl-cc/unifiedconfig/config/precisionqueue/5000 Content-Type: text/plain Content-Length: 0 Date: Tue, 12 Jan 2010 16:00:00 GMT </pre> <p>See also HTTP Responses, on page 7.</p>

**Operation
Validation:****General Restrictions**

- You can create a maximum of 2000 precision queues to the entire system.
- You can create a maximum of 5000 precision queue steps to the entire system.
- You can associate a maximum of ten precision queue terms to a precision queue step.
- You can associate a maximum of five Attributes to a precision queue term.

For Precision Queue Parameters:

bucketInterval

- Optional field.

agentOrdering

- Required field.
- Integer.
- Valid values are 1, 2, and 3.

callOrdering

- Required field.
- Integer.
- Valid values is: 1.

name

- Required field.
- Max length is 32 bytes.

description

- Optional field.
- Max length is 255 bytes.
- For details on valid characters for this field, see [Internationalization](#), on page 9.

serviceLevelThreshold

- Required field.
- Integer.

serviceLevelType

- Required field.
- Integer.
- Valid values are: 1, 2, and 3.

steps

- Required field.

For bucketInterval Parameters:

name

- Required field.

refURL

- Required field.

For step Parameters:

stepOrder

- Required field.
- Integer.
- This value must start at 1 (0 is invalid) and increment by one for each subsequent step.

waitTime

- Required field.
- Integer.
- The value must be 0 or greater for all steps, excluding the last step. The value for the last step defaults to -1. With a value of -1, the system waits until an Agent is available to take the call.

considerIf

- Required field.
- Max length 255 bytes.
- Objects used in the expression are case sensitive.
- You cannot add an expression to the last step.
- Examples:
 - `PQ.PQ1.LoggedOn > 1`—This expression evaluates whether there is more than one Agent logged into this queue.
 - `CallType.CallType1.CallsRoutedToday > 100`—Evaluates whether more than 100 calls of this Call Type were routed today.

description

- Optional field.

terms

- Required field.

For term Parameters:

termOrder

- Required field.
- Integer.
- Value must start at 1 (zero is invalid) for each new step and increment by one for each subsequent term.

attribute

- Required field.
- All elements inside the Attribute reference node, except the refURL element, are read-only.

parenCount

- Required field.
- Integer.
- The total of ParenCount for all Attribute entries for a precision queue must be zero.

termRelation

- Required field.
- Integer.
- Valid values are 0, 1, and 2.

attributeRelation

- Required field.
- Integer.
- Valid values are 1-6.

value1

- Required field.
- Must be convertible to the data type specified in the Attribute table.

delete

Deletes a precision queue and the corresponding precision queue steps and terms.

Note: If you reference a precision queue statically in any version of a saved script, you cannot delete it from the precision queue. Specifically, to delete a precision queue that is referenced statically in a script, you must delete it from every version of the saved script.

If you reference a precision queue dynamically in a script and there are calls queued against the precision queue, you can delete the precision queue. However, the router keeps the precision queue operational until all calls are no longer in the queue. Once the precision queue is deleted, no new calls are queued against it.

URL:	https://<server>:port/unifiedconfig/config/precisionqueue/<id>
HTTP Method:	DELETE
HTTP Response Headers:	HTTP/1.1 200 OK Content-Type: text/plain Content-Length: 0 Date: Tue, 12 Jan 2010 16:00:00 GMT See also HTTP Responses , on page 7.

list

Use the LIST API to retrieve all existing precision queues and optionally agent count and attributes.

agentcount	True/False	An optional parameter that returns agent count for each returned precision queue and each step.
attributes	True/False	An optional parameter that returns a list of attributes for each returned precision queue.

Examples:

- **URL:** https://<server>/unifiedconfig/config/precisionqueue?agentcount=true
- **URL:** https://<server>/unifiedconfig/config/precisionqueue?attributes=true
- **URL:** https://<server>/unifiedconfig/config/precisionqueue?agentcount=true&attributes=true

URL:	https://<server>/unifiedconfig/config/precisionqueue
HTTP Method:	GET
Summary Parameter URL:	https://<server>/unifiedconfig/config/precisionqueue?summary=2

Example XML Response:	<pre> <precisionQueues> <precisionQueue> <changeStamp>3</changeStamp> <refURL>https://<server>/unifiedconfig/config/precisionqueue/5007</refURL> <agentOrdering>1</agentOrdering> <callOrdering>1</callOrdering> <description>sales precision queue</description> <name>MarketingAB</name> <serviceLevelThreshold>1</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> </precisionQueue> <precisionQueue> <changeStamp>3</changeStamp> <refURL>https://<server>/unifiedconfig/config/precisionqueue/5008</refURL> <agentOrdering>1</agentOrdering> <callOrdering>1</callOrdering> <description>sales precision queue</description> <name>MarketingABC</name> <serviceLevelThreshold>1</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> </precisionQueue> <precisionQueue> <changeStamp>3</changeStamp> <refURL>https://<server>/unifiedconfig/config/precisionqueue/5012</refURL> <agentOrdering>1</agentOrdering> <callOrdering>1</callOrdering> <description>sales precision queue</description> <name>Sales225</name> <serviceLevelThreshold>1</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> </precisionQueue> </precisionQueues> </pre>
HTTP Response Headers:	<pre> HTTP/1.1 200 OK Content-Type: application/xml Transfer-Encoding: chunked Date: Tue, 12 Jan 2010 16:00:00 GMT </pre>

get

Use to retrieve a precision queue, the corresponding steps and terms, and optionally agent count and attributes.

agentcount	True/False	An optional parameter that returns agent count for the precision queue and each step.
attributes	True/False	An optional parameter that returns a list of attributes for the precision queue.

Examples:

- URL: `https://<server>/unifiedconfig/config/precisionqueue/id?agentcount=true`

- URL: `https://<server>/unifiedconfig/config/precisionqueue/id?attributes=true`
- URL: `https://<server>/unifiedconfig/config/precisionqueue/id?agentcount=true&attributes=true`

URL:	<code>https://<server>:port/unifiedconfig/config/precisionqueue/<id></code>
HTTP Method:	GET
Example XML Response:	<pre> <precisionQueue> <changeStamp>4</changeStamp> <refURL>/unifiedconfig/config/precisionqueue/5002</refURL> <agentOrdering>1</agentOrdering> <bucketInterval> <refURL>/unifiedconfig/config/bucketinterval/1</refURL> <name>Default_Bucket_Intervals</name> </bucketInterval> <callOrdering>1</callOrdering> <description>This is a practice precision queue</description> <name>Practice_Queue</name> <serviceLevelThreshold>3</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> <steps> <step> <terms> <term> <attribute> <refURL>/unifiedconfig/config/attribute/5698</refURL> <name>test</name> <dataType>4</dataType> </attribute> <attributeRelation>5</attributeRelation> <parenCount>0</parenCount> <termRelation>0</termRelation> <value1>2</value1> </term> </terms> <waitTime>0</waitTime> </step> <step> <terms> <term> <attribute> <refURL>/unifiedconfig/config/attribute/5698</refURL> <name>test</name> <dataType>4</dataType> </attribute> <attributeRelation>3</attributeRelation> <parenCount>0</parenCount> <termRelation>0</termRelation> <value1>2</value1> </term> </terms> <waitTime>-1</waitTime> </step> </steps> </precisionQueue> </pre> <p>See HTTP Responses, on page 7.</p> <p>Note The <id> is the precision queue database ID (precisionQueueID). It is a read-only field, shown only in the list and get API commands.</p>
Operation Validation:	<p>agentCount</p> <ul style="list-style-type: none"> • Optional field.

	<ul style="list-style-type: none"> • Integer.
agentCount Parameter URL:	https://<server>/unifiedconfig/config/precisionqueue/<id>?agentcount=true

update

Updates a precision queue and corresponding steps and terms.

When you update an existing precision queue, you must provide all of the parameters. There are no incremental updates of any parameters, including steps and terms. For example, assume you have a precision queue with a single step and a single term. To add a term to the existing step, you must provide all precision queue parameters, in addition to the step and both terms.

When you update an existing precision queue, Agents associated with the queue can change dynamically. Specifically, if you modify the terms associated with precision queue steps, Agents that were previously in the precision queue may be removed from the queue and Agents that were previously not included in the precision queue may be added to the queue. For example, you have a precision queue with one step and one term and the term criterion is: Spanish > 6. If Agent A has the attribute Spanish with value 5, the Agent is not included in this precision queue. If you update the precision queue and change the term to Spanish >= 5, Agent A is dynamically moved into this precision queue and is available to take calls.

URL:	https://<server>:port/unifiedconfig/config/precisionqueue/<id>
HTTP Method:	PUT
Input/Output Format:	xml
Parameters:	See API Parameters , on page 133.

XML Request Payload:	<pre> <precisionQueue> <bucketInterval> <name>PQBucket1</name> <refURL>http://localhost/unifiedconfig/config/bucketinterval/5001</refURL> </bucketInterval> <agentOrdering>1</agentOrdering> <callOrdering>1</callOrdering> <description>sales precision queue</description> <name>SalesPQ</name> <serviceLevelThreshold>1</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> <changeStamp>3</changeStamp> <steps> <step> <stepOrder>1</stepOrder> <waitTime>-1</waitTime> <considerIf>1</considerIf> <nextStep>>false</nextStep> <description>"Step 1"</description> <terms> <term> <attributeURL>http://localhost/unifiedconfig/config/attribute/5032</attributeURL> <termOrder>1</termOrder> <parenCount>1</parenCount> <termRelation>0</termRelation> <attributeRelation>5</attributeRelation> <value1>2</value1> </term> <term> <attributeURL>http://localhost/unifiedconfig/config/attribute/5032</attributeURL> <termOrder>2</termOrder> <parenCount>-1</parenCount> <termRelation>1</termRelation> <attributeRelation>3</attributeRelation> <value1>5</value1> </term> </terms> </step> </steps> </precisionQueue> </pre>
Operation Validation:	<p>Restrictions:</p> <p>All of the restrictions from the create operation apply, in addition to the following:</p> <p>changeStamp</p> <ul style="list-style-type: none"> • Integer. • The same value must be returned in an update operation, to ensure the client is working with the latest data set. This value is not required when you create a precision queue.



Reason Code API

You can use the Reason Code API allows to define new Reason Codes, and edit or delete records of existing Reason Codes.

- [Reason Code API Commands, page 147](#)

Reason Code API Commands

This section explains the five supported API operations for Reason Code and their parameters.

API Parameters

The following table shows the parameters for Reason Code API:

Parameter Name	Description
text	The text that describes the Reason Code.
code	The Reason Code.
description	Additional information about the Reason Code.
changeStamp	Incremented when the record is changed in the database.

create

Creates one Reason Code and stores it in the database.



Note

The Reason Code must be unique in the database. The ReasonCodeID is generated by the configuration application.

URL:	https://<server>/unifiedconfig/config/reasoncode
HTTP Method:	POST
Parameters:	See API Parameters , on page 147.
Example XML Request Payload:	<pre><reasonCode> <text>Example reason text</text> <code>12345</code> <description>example description</description> <changeStamp>1</changeStamp> </reasonCode></pre>
Operation Validation:	<p>text</p> <ul style="list-style-type: none"> • Required field. • Max length of 40 bytes allowed. • Internationalized characters are not allowed. <p>code</p> <ul style="list-style-type: none"> • Required field. • Cannot be modified. • Positive Integers between 0 and maximum: 65535. Must also be unique. • Value cannot be updated once entered. <p>Note To change the code value, the Reason Code must be deleted and then re-added.</p> <p>description</p> <ul style="list-style-type: none"> • Optional field. • Max length of 255 bytes allowed. • For details on valid characters for this field, see Internationalization, on page 9. <p>changeStamp</p> <ul style="list-style-type: none"> • Initial value 0. • Not applicable for Create Operations.

delete

Deletes one Reason Code from the database.



Note

The delete only *marks* the record for deletion, it does not permanently delete the Reason Code.

URL:	https://<server>/unifiedconfig/config/reasoncode/<id>
-------------	---

HTTP Method:	DELETE
Response:	See also HTTP Responses , on page 7.

list

Retrieves a list of Reason Codes.

URL:	https://<server>/unifiedconfig/config/reasoncode
HTTP Method:	GET
Response:	See Permissions Information , on page 12. See also HTTP Responses , on page 7.

get

Returns one Reason Code from the database.

URL:	https://<server>/unifiedconfig/config/reasoncode/<id>
HTTP Method:	GET
Response:	See also HTTP Responses , on page 7.

update

Updates one Reason Code in the database.



Note

You can change the **text** and **description** fields. However, you must permanently delete the Reason Code and recreate a new one to reuse the value in the code field.

URL:	https://<server>/unifiedconfig/config/reasoncode/<id>
HTTP Method:	PUT
Example XML Request Payload:	<pre><reasonCode> <text>Example reason text</text> <code>12345</code> <description>example description</description> <changeStamp>1</changeStamp> </reasonCode></pre>
Response:	See HTTP Responses , on page 7.

Asynchronous API

See section on [Asynchronous API](#), on page 20.



Note

For Reason Code API, the Asynchronous feature is supported only for the create, update, and delete operations.



Serviceability API

You can use the Serviceability API to view Serviceability information.

- [Serviceability API Commands, page 151](#)
- [Serviceability Categories, page 158](#)

Serviceability API Commands

This section provides an introduction to Serviceability API and explains the supported API operations for Serviceability and outlines their parameters.

Introduction

The Serviceability API is used to return information from the system, such as API statistics, version information, and selected JMX statistics.

The following query returns two items: *system info summary* and *version* information:

```
https://<server>/unifiedconfig/config/serviceability?category=systemInfoSummary&category=version
```

The following query returns all information except the **<systemInfo>** element.

```
https://<server>/unifiedconfig/config/serviceability
```

API Parameters

The following table shows the parameters for Serviceability API:

Parameter Name	Description
category	The type of data to return.

The following table shows the possible values for **category**. Note that multiple **category** elements may be supplied on the query string.

Category Value	Description
version	Version information for Packaged CCE.
ucceVersion	Version information for Unified CCE itself.
systemInfo	All JMX MBean values.
systemInfoSummary	Selected JMX MBean values (ones that are most relevant for the purpose of monitoring performance).
jvmStats	JVM statistics (for example: CPU and heap percentage samples, uptime).
apiStats	API statistics (for example: last time, max time, average time, request count).
capacityInfo	See Capacity , on page 158.
all	To return every category, including "systemInfo".
(not supplied)	To show everything except "systemInfo".
systemValidationStatus	To return overall system validation status and individual rules status.

**Note**

For a quicker response from the web application, the **systemInfo** category is excluded, unless explicitly requested.

get

Returns Serviceability information.

URL:	https://<server>/unifiedconfig/config/serviceability?category=<cat1>&category=<cat2>
HTTP Method:	GET
Parameters:	See API Parameters , on page 151.

Response:

```
<Serviceability>
  <currentTime>Tue Nov 29 04:00:45 EST 2011</currentTime>
  <instanceName>instance</instanceName>
  <version>
    <majorVersion>9</majorVersion>
    <minorVersion>0</minorVersion>
    <maintenanceVersion>0</maintenanceVersion>
```

```

    <srVersion>0</srVersion>
    <esVersion>0</esVersion>
    <buildVersion>1</buildVersion>
    <buildDate>1969-12-31T19:00:00-05:00</buildDate>
  </version>
  <ucceVersion>
    <majorVersion>9</majorVersion>
    <minorVersion>5</minorVersion>
    <maintenanceVersion>3</maintenanceVersion>
    <srVersion>0</srVersion>
    <esVersion>0</esVersion>
    <buildVersion>375</buildVersion>
    <versionString>9.5.3.0.0.375</versionString>
    <patchInfos>
      <patchInfo>
        <majorVersion>9</majorVersion>
        <minorVersion>5</minorVersion>
        <maintenanceVersion>3</maintenanceVersion>
        <srVersion>0</srVersion>
        <esVersion>0</esVersion>
      </patchInfo>
      <patchInfo>
        <majorVersion>9</majorVersion>
        <minorVersion>5</minorVersion>
        <maintenanceVersion>2</maintenanceVersion>
        <srVersion>0</srVersion>
        <esVersion>0</esVersion>
      </patchInfo>
      <patchInfo>
        <majorVersion>9</majorVersion>
        <minorVersion>0</minorVersion>
        <maintenanceVersion>1</maintenanceVersion>
        <srVersion>0</srVersion>
        <esVersion>0</esVersion>
      </patchInfo>
    </patchInfos>
  </ucceVersion>
  <apiStats>
    <apiStats>
      <ApiInfo>
        <apiName>serviceability-GET</apiName>
        <requestAverageTime>146</requestAverageTime>
        <requestLastTime>77</requestLastTime>
        <requestLongestTime>511</requestLongestTime>
        <requestShortestTime>12</requestShortestTime>
        <requestsCount>6</requestsCount>
        <requestsLastInterval>0</requestsLastInterval>
        <requestsTotalTime>878</requestsTotalTime>
        <tooLongLastInterval>0</tooLongLastInterval>
        <tooLongTotal>0</tooLongTotal>
      </ApiInfo>
    </apiStats>
  </apiStats>
  <jvmStats>
    <jmxServers>
      <jmxServer>
        <connectionName>Runtime - local</connectionName>
        <cpuPercentageSamples>4.2,0.4,0.5,0.4</cpuPercentageSamples>
        <heapSamples>24,19,22</heapSamples>
        <heapSlope>-7.397258766251369E-6</heapSlope>
        <uptime xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:long">61314179</uptime>
        <uptimeString>17 hrs 01 min 54 sec</uptimeString>
      </jmxServer>
    </jmxServers>
  </jvmStats>
  <systemInfoSummary>
    <jmxInfo>
      <JmxServer>
        <connectionName>Runtime - local</connectionName>
        <mbeanValues>
          <mbeanValue>

```

```

<name>Catalina:J2EEApplication=none,J2EEServer=none,WebModule=//localhost/unifiedconfig,j2eeType=Servlet,name=REST.errorCount</name>
    <value>0</value>
  </mbeanValue>
</mbeanValues>
</jmxInfo>
</systemInfoSummary>
</Serviceability>

```

The following table shows the Response Elements.

Element	Description
currentTime	The time at which this web request was made.
instanceName	The name of the active Unified CCE instance.
buildDate	The date the application was built.
buildVersion	The build number of the application.
esVersion	The engineering special (ES) version.
maintenanceVersion	The maintenance version.
majorVersion	The major version.
minorVersion	The minor version.
srVersion	The SR version.
versionString	Textual representation of the Unified CCE version.

Exceptions: none.

System Configuration Validation Rules

The System Configuration Validation API validates whether a system meets product specifications. The following rules define a Cisco Packaged Contact Center Enterprise system:

- One Generic PG (ClientType=42) exists with 4 VRU PIMs (ClientType=13) and 1 UCM PIM (ClientType=30).
- The only other PG is a single MR PG (ClientType = 47) with one or more MR PIMs (optional, if Outbound is enabled).
- Single Type 10 Network VRU exists and has the 4 VRU PIMS.

The validation is done with the following restrictions:

- Missing or additionally configured elements flag the system as a non-Packaged CCE system.
- The API allows the user to validate the system as a whole, based on the preceding rules, not based on *individual* rules.

System Configuration Validation Output

The `get` operation returns the system validation status.

URL:	https://<server>/unifiedconfig/config/serviceability?category=systemValidationStatus
HTTP Method:	GET
Parameters:	System Configuration Validation is done using the API Parameters, on page 151 , with the category value <code>systemValidationStatus</code> , which returns overall system validation status and individual rules status.

Successful Validation Scenario

System has the following configuration:

- One Generic PG with 4 VRU PIMs, each associated with a Type10 Network VRU and 1 UCM PIM.
- One MR PG with 1 MR PIM.
- Each rule listed below maps to an enum that has an explanation of the rule in the message bundle:
 - `ECC_VARIABLES_CTI_SIZE` — ECC Variables: Total bytes required for enabled variables in CTI Server must not exceed 2500.
 - `CMS_NODE_DISABLED` — CMS Node: Configuration Management Service (CMS) Node and Agent Re-skilling Web Tool must be disabled using Unified CCE Web Setup.
 - `ENTERPRISE_SERVICE_COUNT` — Enterprise Services: No Enterprise Services may be configured.
 - `MR_PIM_COUNT` — Peripheral: Exactly 2 MR Peripherals must be configured on the MR PG in the PG Explorer tool.
 - `DESK_SETTING_WITH_RING_NO_ANSWER_SET_COUNT` — Agent Desk Settings: Ring No Answer Times must not be set.
 - `ENT_SG_COUNT` — Skill Groups: A maximum of 1 Enterprise Skill Group can be configured.
 - `ENT_SG_MEMBER_COUNT` — Skill Groups: No Enterprise Skill Group Members may be configured.
 - `ENT_ROUTE_COUNT` — Enterprise Routes: A maximum of 1 Enterprise Route must be configured.
 - `ENT_ROUTE_MEMBER_COUNT` — Enterprise Routes: No Enterprise Route Members may be configured.
 - `GENERIC_PG_COUNT` — Peripheral Gateway: Exactly 1 Generic Peripheral Gateway must be configured.
 - `MR_PG_COUNT` — Peripheral Gateway: Exactly 1 Media Routing Peripheral Gateway must be configured.
 - `MULTICHANNEL_COUNT` — Peripheral: Exactly 1 MR PIM must be configured with the Enterprise Name of **Multichannel**.

- **OUTBOUND_COUNT** — Peripheral: Exactly 1 MR PIM must be configured with the Enterprise Name of **Outbound**.
- **PG_COUNT** — Peripheral Gateway: Exactly 2 Peripheral Gateways must be configured.
- **SERVICE_MEMBER_COUNT** — Service Members: No Service Members may be configured.
- **TYPE10_NETWORK_VRU_COUNT** — VRU: Exactly 1 Type 10 Network VRU must be configured in the Network VRU Explorer.
- **TYPE10_NETWORK_VRU_MAP_COUNT** — Peripheral Gateway: All 4 VRU Peripherals must be configured on the Generic PG and associated with the Type 10 Network VRU.
- **UCM_PIM_COUNT** — Peripheral: Exactly 1 Unified CM Peripheral must be configured on the Generic PG in the Peripheral Explorer tool.
- **VRU_PIM_COUNT** — Peripheral: Exactly 4 VRU Peripherals must be configured.
- **NOT_SKILL_GROUP_ROUTE_NAME_COUNT** — Skill Groups: All Skill Group records must have a corresponding Route record with the same Enterprise Name as the Skill Group record.
- **ECC_VARIABLES_ENABLED_COUNT** — ECC Variables: ECC variables must be enabled in the System Information tool.
- **SERVICE_COUNT** — Services: No Services may be configured.
- **TRANSLATION_ROUTE_COUNT** — Translation Routes: No Translation Routes may be configured.
- **NIC_COUNT** — NICs: No NICs may be configured.
- **MRD_COUNT** — Max Media Routing Domains: The maximum number of Media Routing Domains is 20.
- **MEDIA_CLASS_COUNT** — Max Media Classes: The maximum number of Media Classes is 10.
- **NOT_PARTITIONED_COUNT** — Partitioning: Partitioning must be disabled in the System Information tool.
- **NON_NULL_SERVICE_LEVEL_COUNT** — Service Level Threshold: The default service level must not be set in the Peripheral Explorer tool. The default is set in the System Information tool.
- **DEVICE_TARGET_COUNT** — Device Targets: No Device Targets can be configured.
- **CVP_LABEL_COUNT** — VRU: Each VRU PIM associated with the Generic PG in the PG Explorer tool must have exactly 1 label with a length of 10 digits.
- **CUCM_LABEL_COUNT** — CUCM Routing Label: Exactly 1 label with length of 10 digits must be configured and associated with the Unified CM routing client.
- **CORRELATION_ID_RANGE_COUNT** — Correlation ID: The minimum and maximum correlation number in the VRU section of the System Information tool must be 1001 and 9999 respectively.
- **NULL_FEATURE_SET_ID_COUNT** — Feature Control Set : The Feature Control Set in the Customer Definition of the ICM Instance Explorer must set to NONE.
- **ECC_FOR_CVP_COUNT** — ECC Variables: Exactly 9 Expanded Call Variables are required for CVP.

- NETWORK_VRU_SCRIPT_COUNT — VRU: There must be a Network VRU Script with the Enterprise Name of **VXML_Server** and the Script Name of **GS,Server,V** configured in the Network VRU Script tool.
- DEFAULT_DESK_SETTING_COUNT — Agent Desk Settings: Default_Agent_Desk_Setting must be set as the default Agent Desk Settings for the CUCM PIM in the PG Explorer tool.
- PCCE_APP_INSTANCE_MULTICHANNEL_COUNT — Multichannel Application Instance: An Application Instance must be defined for Multichannel.
- TYPE2_NETWORK_VRU_COUNT — VRU: Exactly one Type 2 Network VRU must be configured in the Network VRU Explorer tool.
- TYPE2_NETWORK_VRU_MAP_COUNT — Peripheral: Both MR PIMs must be associated with a Type 2 Network VRU in the PG Explorer tool.
- DIALED_NUMBER_EXTERNALL_VOICE_COUNT — Dialed Numbers: For each External Voice Dialed Numbers, there must be exactly 4 Dialed Number records for each Dialed Number String, with one for each VRU PIM.
- DIALED_NUMBER_MAP_COUNT — Dialed Numbers: All Dialed Number records must not have an associated Region, ANI, and must have a maximum of 1 Call Type associated in the Call Type Map.
- AGENT_REAL_TIME_ENABLED_COUNT — Peripheral: Agent Reporting must be enabled on the Unified CM Peripheral in the PG Explorer tool.
- CUSTOMER_DEFINITION_COUNT - Customer Definition: Exactly 1 Customer Definition must be configured in the ICM Instance Explorer.
- CUSTOMER_DEFINITION_HAS_TYPE10_NETWORK_VRU - Customer Definition: Exactly 1 Customer Definition must have a Type 10 Network VRU selected in the ICM Instance Explorer.
- DIALED_NUMBERS_REQUIRE_CUSTOMER_DEFINITION - Dialed Numbers: No Dialed Number records can have the Customer set to None.
- SCRIPT_VERSIONS_TO_RETAIN - Script Versions to Retain: The number of script versions to retain must be between 1 and 100, inclusively.
- APP_GATEWAY_COUNT - Application Gateway: No Application Gateways may be configured.
- DATABASE_LOOKUP_COUNT - Database Lookups: No Database Lookups may be configured.
- TEMPDEV_AUTOGROWTH - Autogrowth: The tempdev file in the tempdb database must have autogrowth enabled and set, in percent, to 10.
- TEMPLOG_AUTOGROWTH - Autogrowth: The templog file in the tempdb database must have autogrowth enabled and set, in percent, to 10.

Response:

```
<Serviceability>
  <currentTime>Tue Nov 29 04:00:45 EST 2011</currentTime>
  <systemValidationStatus>
    <isValid>true</isValid>
    <validationRules>
      <validationRule>
        <isValid>true</isValid>
        <name>PG_COUNT</name>
        <min>1</min>
        <max>2</max>
```

```

        <actual>2</actual>
      </validationRule>
    <validationRule>
<isValid>true</isValid>
<name>GENERIC_PG_COUNT</name>
    <min>1</min>
    <max>1</max>
    <actual>1</actual>
  </validationRule>
  <validationRule>
<isValid>true</isValid>
<name>VRU_PIM_COUNT</name>
    <min>4</min>
    <max>4</max>
    <actual>4</actual>
  </validationRule>
  <validationRule>
<isValid>true</isValid>
<name>TYPE10_NETWORK_VRU_COUNT</name>
    <min>1</min>
    <max>1</max>
    <actual>1</actual>
  </validationRule>
  <validationRule>
<isValid>true</isValid>
<name>TYPE10_NETWORK_VRU_MAP_COUNT</name>
    <min>4</min>
    <max>4</max>
    <actual>4</actual>
  </validationRule>
  <validationRule>
<isValid>true</isValid>
<name>UCM_PIM_COUNT</name>
    <min>1</min>
    <max>1</max>
    <actual>1</actual>
  </validationRule>
  <validationRule>
<isValid>true</isValid>
<name>MR_PG_COUNT</name>
    <min>0</min>
    <max>1</max>
    <actual>1</actual>
  </validationRule>
</validationRules>
</systemValidationStatus>
</Serviceability>

```

Serviceability Categories

This section explains the Serviceability Categories.

Capacity

The Serviceability API has a **capacity** category, which shows capacity information. For this category, the **get** operation returns information on the current capacity limits.

URL:	<a href="https://<server>/unifiedconfig/config/serviceability?category=capacityInfo">https://<server>/unifiedconfig/config/serviceability?category=capacityInfo
HTTP Method:	GET

Response:

```
<serviceability>
  <currentTime>Tue Nov 29 04:00:45 EST 2011</currentTime>
  <capacityInfo>
    <capacityRules>
      <capacityRule>
        <name>MAXIMUM_CALL_TYPES</name>
        <maximum>1000</maximum>
        <actual>125</actual>
      </capacityRule>
      <capacityRule>
        <name>MAXIMUM_DIALED_NUMBER_EXTERNAL</name>
        <maximum>1000</maximum>
        <actual>317</actual>
      </capacityRule>
    </capacityRules>
  </capacityInfo>
</serviceability>
```

Note: Each **capacityCheckResult** returns:

- The name of the rule checked.
- The maximum number of items allowed by that rule.
- The current number of items configured.



Skill Group API

You can use the Skill Group API to list the Skill Groups currently defined in the database, define new Skill Groups, and view, edit, or delete records of existing Skill Groups.

- [Skill Group API Commands, page 161](#)

Skill Group API Commands

This section explains the five supported API operations for Skill Group and outlines the parameters.

API Parameters

The following table shows the parameters for Skill Group API:

Parameter Name	Description
name	The name of the Skill Group. This name must be unique.
description	Additional information about the Skill Group.
changeStamp	Incremented when the record is changed in the database (read-only).
mediaRoutingDomain	Identifier for the Media Routing Domain.
agents	A collection of Agents assigned to the Skill Group.
bucketInterval	Identifier of the Bucket Interval level, used for reporting.
serviceLevelThreshold	Maximum time in seconds that a caller should wait before being connected with an Agent.

Parameter Name	Description
serviceLevelType	This value indicates how the system calculates the service level. <ul style="list-style-type: none"> • NULL = Use the system default. • Ignore Abandoned Calls = 1. • Abandoned Calls have Negative Impact = 2. • Abandoned Calls have Positive Impact = 3.
peripheralNumber	Read-only.

create

Creates a Skill Group record and stores the data in the database.

URL:	https://<server>/unifiedconfig/config/skillgroup
HTTP Method:	POST
Input/Output Format:	xml
Parameters:	See API Parameters , on page 161.
Example XML Request Payload:	<pre> <skillGroup> <refURL>/unifiedconfig/config/skillgroup/(id)</refURL> <name>test</name> <description>test skill group</description> <changeStamp>0</changeStamp> <mediaRoutingDomain> <name>Cisco_Voice</name> <refURL>/unifiedconfig/config/mediaroutingdomain/1</refURL> </mediaRoutingDomain> <bucketInterval> <name>bucketIntervalName</name> <refURL>/unifiedconfig/config/bucketinterval/1</refURL> </bucketInterval> <serviceLevelThreshold>20</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> <peripheralNumber>1234567</peripheralNumber> <agents> <agent> <refURL>/unifiedconfig/config/agent/5000</refURL> <firstName>Jane</firstName> <lastName>Doe</lastName> <agentId>8007</agentId> </agent> <agent>...</agent> <agent>...</agent> </agents> </skillGroup> </pre>
Response:	In the Response, the Location header has a URL to the newly created Skill Group, if successful. See also HTTP Responses , on page 7.

	<p>Note For the create operation, the firstName, lastName, and agentId in <agent> are ignored.</p>
<p>Operation Validation:</p>	<p>name</p> <ul style="list-style-type: none"> • Required field. • Max length of 32 bytes allowed. • Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric. • Name must be unique. • Does not allow internationalized characters. <p>description</p> <ul style="list-style-type: none"> • Optional field. • No restriction of characters. • Max length of 255 bytes allowed. • For details on valid characters for this field, see Internationalization, on page 9. <p>changeStamp</p> <ul style="list-style-type: none"> • Optional field. • Integers only: start with 0. • Required for the update operation. <p>mediaRoutingDomain</p> <ul style="list-style-type: none"> • Optional field. Defaults to Cisco_Voice MRD if this field is not provided. • Must be a valid refURL from the Media Routing Domain API. • This field cannot be updated. <p>bucketInterval</p> <ul style="list-style-type: none"> • Optional field. • Must be a valid refURL from the Bucket Interval API. <p>serviceLevelThreshold</p> <ul style="list-style-type: none"> • Optional field. • Positive integers only, or blank. Blank means use the peripheral default. <p>serviceLevelType</p> <ul style="list-style-type: none"> • Values can only be NULL (default), 1, 2, 3. <p>peripheralNumber</p> <ul style="list-style-type: none"> • Read-only field.

- Automatically generated when using the create operation.

Note A Route record is maintained seamlessly by the Skill Group API; that is, a single Route record is generated for each Skill Group created and the process is hidden from the user. The Route stentry is generated with the Skill Group Name and Skill Target ID. The Route records are updated and deleted via the Skill Group API.

delete

Deletes one Skill Group from the database.



Note

The operation only *marks* the record for deletion, it does not permanently delete it.

URL:	https://<server>/unifiedconfig/config/skillgroup/<id>
HTTP Method:	DELETE
Response:	See also HTTP Responses , on page 7.
Operation Validation:	You cannot delete a Skill Group that is referenced by: <ul style="list-style-type: none"> • A script. • A campaign.

list

Retrieves a list of Skill Groups.

URL:	https://<server>/unifiedconfig/config/skillgroup
HTTP Method:	GET

Example XML Response:

```
<results>
  <pageInfo>
    ... (see pagination data)
  </pageInfo>
  <skillGroups>
    <skillGroup>...</skillGroup>
    <skillGroup>...</skillGroup>
  </skillGroups>
</results>
```

Note The preceding example XML response does not show all of the data for pagination. See [Pagination, on page 10](#).

Note Also, the example XML response does not show permissions information. See [Permissions Information, on page 12](#).

See also [HTTP Responses, on page 7](#).

get

Returns one Skill Group record from the database.

URL:	https://<server>/unifiedconfig/config/skillgroup/<id>
HTTP Method:	GET

Example XML Response:

```

<skillGroup>
  <refURL>http://<server>/unifiedconfig/config/skillgroup/(id)</refURL>
  <name>test</name>
  <description>test skill group</description>
  <changeStamp>0</changeStamp>
  <mediaRoutingDomain>
    <name>Cisco_Voice</name>

  <refURL>https://10.86.135.206/unifiedconfig/config/mediaroutingdomain/1</refURL>

  </mediaRoutingDomain>
  <bucketInterval>
    <name>bucketIntervalName</name>

  <refURL>https://10.86.135.206/unifiedconfig/config/bucketinterval/1</refURL>
  </bucketInterval>
  <serviceLevelThreshold>20</serviceLevelThreshold>
  <serviceLevelType>1</serviceLevelType>
  <peripheralNumber>1234567</peripheralNumber>
  <agents>
    <agent>
      <refURL>https://10.86.135.206/unifiedconfig/config/agent/5000</refURL>

      <firstName>Jane</firstName>
      <lastName>Doe</lastName>
      <userName>username</userName>
      <agentId>8007</agentId>
      <canRemove>true</canRemove>
    </agent>
    <agent>
      <refURL>https://10.86.135.206/unifiedconfig/config/agent/5001</refURL>

      <firstName>John</firstName>
      <lastName>Smith</lastName>
      <userName>username2</userName>
      <agentId>8008</agentId>
      <agentTeam>
        <refURL>/unifiedconfig/config/agentteam/5000</refURL>
        <name>someTeam</name>
      </agentTeam>
      <canRemove>false</canRemove>
    </agent>
    <agent>...</agent>
    <agent>...</agent>
  </agents>
</skillGroup>

```

Note The <canRemove> field only appears for Supervisors. It indicates if the Supervisor has permission to remove the Agent from this skill group. The Supervisor can remove the Agent from the skill group if the Agent belongs to a team of this supervisor.

Note The <agentTeam> data is visible only to Agents who are on teams.

See [HTTP Responses](#), on page 7.

update

Updates one Skill Group record in the database.

URL:	https://<server>/unifiedconfig/config/skillgroup/<id>
HTTP Method:	PUT
Input/Output Format:	xml

Example XML Request Payload:	<pre> <skillGroup> <refURL>/unifiedconfig/config/skillgroup/(id)</refURL> <name>test</name> <description>test skill group</description> <changeStamp>0</changeStamp> <mediaRoutingDomain> <name>Cisco_Voice</name> <refURL>/unifiedconfig/config/mediaroutingdomain/1</refURL> </mediaRoutingDomain> <bucketInterval> <name>bucketIntervalName</name> <refURL>/unifiedconfig/config/bucketinterval/1</refURL> </bucketInterval> <serviceLevelThreshold>20</serviceLevelThreshold> <serviceLevelType>1</serviceLevelType> <peripheralNumber>1234567</peripheralNumber> <agents> <agent> <refURL>/unifiedconfig/config/agent/5000</refURL> <firstName>Jane</firstName> <lastName>Doe</lastName> <agentId>8007</agentId> </agent> <agent>...</agent> <agent>...</agent> </agents> </skillGroup> </pre> <p>For the update operation, the firstName, lastName, and agentId in <agent> are ignored, as are the name fields in mediaRoutingDomain and bucketInterval.</p>
Parameters:	See API Parameters , on page 161.
Response:	See HTTP Responses , on page 7.

