Cisco Packaged Contact Center Enterprise Developer Reference, Release 12.0(1)

First Published: 2019-01-11

Last Modified: 2019-11-21

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

CONTENTS

CHAPTER 1 **Working with Cisco Packaged Contact Center Enterprise APIs** 1

- Change Log 1
- API Operations 3
- Access 4
- Usage and Behavior 5
- Error Responses 6
- Pagination 7
- Shared Parameters 8
- Permissions 9
- Synchronous vs. Asynchronous Writes 9
- Search 10
- Sort 11

CHAPTER 2 **Active Directory Domain API** 13

CHAPTER 3 **Administrator API** 15

CHAPTER 4 **Agent API** 17

CHAPTER 5 **Agent Desk Settings API** 25

CHAPTER 6 **Agent State Trace API** 29

CHAPTER 7 **Agent Team API** 31

CHAPTER 8 **Application Gateway API** 35

CHAPTER 9 Application Gateway Global Settings API 41

CHAPTER 10 Attribute API 43

CHAPTER 11 Bucket Interval API 47

CHAPTER 12 Bulk Job API 49

CHAPTER 13 Business Hours API 53

Business Hours API 53

Business Hour Status Reason API 57

Time Zone API 58

CHAPTER 14 Call Type API 61

CHAPTER 15 Congestion Control API 65

CHAPTER 16 Context Service Configuration API 67

CHAPTER 17 Context Service Registration API 69

CHAPTER 18 Courtesy CallBack API 73

CHAPTER 19 CVP Device Configuration API 75

CHAPTER 20 CVP DNIS API 81

CVP Dialed Number Identification Service API 81

CHAPTER 21 CVP Graceful Shutdown 83

CHAPTER 22 Data Center API 85

CHAPTER 23 Department API 89

CHAPTER 24 Deployment API 91

CHAPTER 25 Deployment Type Info API 93

CHAPTER 26 Dialed Number API 95

CHAPTER 27 ECC Payload API 101

CHAPTER 28 Expanded Call Variable API 105

CHAPTER 29 Facility API 109

CHAPTER 30 Global API 111

CHAPTER 31 Initialize API 117

CHAPTER 32 Instance API 119

CHAPTER 33 Internet Script Editor API 121

CHAPTER 34 Inventory Import API 123

Templates 125

CHAPTER 35 Location API 127

Location Properties API 129

CHAPTER 36 Log Collection API 131

CHAPTER 37 Machine Inventory API 133

CHAPTER 38 Media Routing Domain API 143

CHAPTER 39 Network VRU Script API 147

CHAPTER 40 Operation API 149

CHAPTER 41 Outbound API 153

Outbound Campaign API 154

Campaign Status API 161

Do Not Call API 163

Create Do Not Call List 164

Import API 165

Personal Callback API 170

Time Zone API 175

CHAPTER 42 Peripheral Gateway API 177

CHAPTER 43 Precision Queue API 179

CHAPTER 44 Reason Code API 183

CHAPTER 45 Role API 187

CHAPTER 46 Routing Type API 191

CHAPTER 47 Routing Pattern API 193

CHAPTER 48 Scan API 197

CHAPTER 49 Serviceability API 199

System Validation Rules 201

CHAPTER 50 Single Sign-On Global State API 205

CHAPTER 51 Single Sign-On Registration API 207

CHAPTER 52 SIP Server Group API 209

CHAPTER 53 SIP Server Group Properties API 213

CHAPTER 54 Single Sign-On Status API 217

CHAPTER 55 Skill Group API 221

CHAPTER 56 Stats API 227

CHAPTER 57 Status API 229

Configuration Rules 230

Operation Rules 237

System Health Rules 244

VM Rules 245

CHAPTER 58 Trace Level API 247

CHAPTER 59 Transferable Files API 249

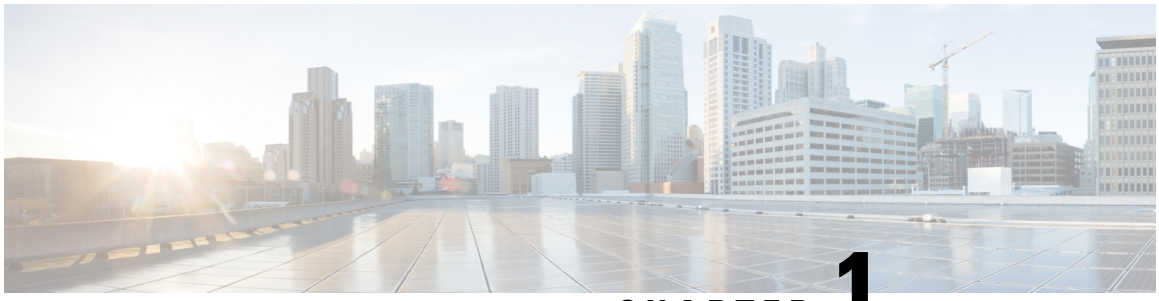
CHAPTER 60 Uninitialize API 251

CHAPTER 61 Version API 253

CHAPTER 62 Virtualized Voice Browser Configuration API 255

APPENDIX A Reference 261

Third-party Integration UI Specification 261



CHAPTER 1

Working with Cisco Packaged Contact Center Enterprise APIs

Cisco Packaged Contact Center Enterprise (Packaged CCE) uses REST-based API functions accessed over HTTP. Each API Operation is mapped to an HTTP method. For more information, see API Operations, on page 3.

This document explains the operations and parameters for each configurable item in Packaged CCE.

For more information on managing the system, see the *Cisco Packaged Contact Center Enterprise Administration and Configuration Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html.

To review examples on how to interact with the Unified Config and Outbound REST web services in Java, see the CCE Config Sample REST Toolkit at https://developer.cisco.com/site/packaged-contact-center/documentation/.

- Change Log, on page 1
- API Operations, on page 3
- Access, on page 4
- Usage and Behavior, on page 5
- Error Responses, on page 6
- Pagination, on page 7
- Shared Parameters, on page 8
- Permissions, on page 9
- Synchronous vs. Asynchronous Writes, on page 9
- Search, on page 10
- Sort, on page 11

Change Log

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

New APIs

- Business Hours API, on page 53
- Courtesy CallBack API , on page 73

- CVP Device Configuration API, on page 75
- CVP Graceful Shutdown, on page 83
- ECC Payload API, on page 101
- Inventory Import API, on page 123
- Location API, on page 127
- Routing Pattern API, on page 193
- SIP Server Group API, on page 209
- SIP Server Group Properties API, on page 213
- Virtualized Voice Browser Configuration API, on page 255

Updated APIs

See	Notes
Machine Inventory API	Updated pairing, username, and type parameter to add FTP_Credential details.
Deployment Type Info API	New hardwareLayoutType parameter added.
Dialed Number API	New pcsPattern, ringtoneName, and peripheralSet parameters added.
Outbound Campaign API	New sort parameters added.
Import API	Added a get operation to obtain the CSV file for the contacts of the campaign.
Data Center API	Added information about the CVP Reporting Server, Cisco Unified SIP Proxy, Cisco Virtualized Voice Browser and Gateway external machines support for both main site and remote sites.
Department API	Added information for Enterprise Chat and Email (ECE).
Machine Inventory API	Added the CVP Reporting Server, Cisco Unified SIP Proxy, Cisco Virtualized Voice Browser and Gateway external machines support for both main site and remote sites in the machine type table. Added a note for the delete operation specific to external machines.
Agent API	Added notes for create, update and delete operations when the supervisor attribute is set to true. The notes are specific to Packaged CCE and Unified Intelligence Center integration. Added new peripheralSet parameter.
Agent Team API	Added notes for create, update and delete operations for Packaged CCE and Unified Intelligence Center integration. Added new peripheralSet parameter.

See	Notes
Skill Group API	Added new peripheralSet parameter.

API Operations

There are five API operations, and they are invoked by HTTP methods.

Responses are provided using HTTP headers and HTTP body containing XML. For information on XML, see XML, on page 6.

create

The create operation uses the HTTP POST method to make one new item and return the URL of that item in the HTTP location header. That URL can then be used to perform the get, update, and delete operations. An XML body containing the parameters and values for the new item must be specified.

delete

The delete operation uses the HTTP DELETE method to delete one item. The item may be marked for deletion or permanently deleted depending on the item type.

To delete more than one item at a time, refer to the Operation API.

You cannot delete **BuiltIn** items (those automatically created by the system, such as the **BuiltIn** bucket interval), items referenced in scripts, or items referenced by other items.

get

The get operation uses the HTTP GET method to retrieve one item. For example, to return one bucket interval record, perform the get operation using the URL:
`https://<server>/unifiedconfig/config/bucketinterval/<id>` .

list

The list operation uses the HTTP GET method to retrieve a list of items. For example, to retrieve a list of bucket intervals, perform the list operation using the URL:
`https://<server>/unifiedconfig/config/bucketinterval`. See also Permissions, on page 9, Pagination, on page 7, Search, on page 10, and Sort, on page 11.

Query parameters:

- Summary list: Some APIs have parameters that include a large amount of data when returned, such as collections of references. Use this query parameter to reduce the number of parameters returned for each item in the list. For example, in the Skill Group API, if skill groups contain a large number of agents, a large amount of data may be returned. Use this query option to return the basic skill group data along with the number of agents having the skill. Append the query parameter `summary=true` to the URL for the API; for example,
`https://<server>/unifiedconfig/config/skillgroup?summary=true`.

update

The update operation uses the HTTP PUT method to modify one item. An XML body containing the parameters and values to update must be specified. For example, to update the name of a bucket interval, perform the update operation on the URL `https://<server>/unifiedconfig/config/bucketinterval/(id)` with the following body:

```
<bucketInterval>
   <name>newName</name>
   <changeStamp>0</changeStamp>
</bucketInterval>
```

Access

Administrator Access

Administrator access to Packaged CCE Administration APIs and items is defined by the role and by the departments for which the administrator is responsible.

Supervisor Access

The following APIs are read only:

- Agent Team API, on page 31
- Precision Queue API, on page 179

The following APIs allow update with restrictions:

- Attribute API, on page 43
  - When updating an attribute, supervisors can only modify the collection of agentAttributes by adding, removing, or changing the value of agents who are on their teams.
- Agent API, on page 17
  - Supervisors can only see and update agents who are on their teams.
  - When updating an agent, supervisors can only change the following parameters:
    - skillGroups
    - defaultSkillGroup
    - skillGroupsAdded
    - skillGroupsRemoved
    - agentAttributes
    - agentAttributesAdded
    - agentAttributesRemoved
    - password
  - The Operation API can also be used to perform updates on agents.
- Skill Group API, on page 221:
  - When updating a skill group, supervisors can only modify the collections of agents by adding or removing agents who are on their teams.

Agent Access

Agents have no access to the Packaged CCE Administration APIs.

Authentication

To authenticate administrators and supervisors must provide a fully qualified user name (for example, user@domain.com) and password.

Usage and Behavior

Duplicate Parameters

If a parameter is duplicated, the final value that is specified will be used by the API.

Read-Only Fields

Read-only parameters are ignored on create and update operations.

References

References are a type of parameter that provide a way to connect one item to another item, defining the relationship between them.

For example, to define which team an agent belongs to, the agent contains a reference to a team. When performing list or get operations, the reference contains the refURL of the item and the name. For example:

```
<agent>
   <team>
     <refURL>/unifiedconfig/team/5000</refURL>
     <name>NameOfTeam</name>
   </team>
  …
</agent>
```

For items that do not have a name parameter, other parameters such as firstName and lastName are included.

```
<agent>
  <refURL>/unifiedconfig/config/agent/5000</refURL>
  <firstName>Jane</firstName>
  <lastName>Doe</lastName>
  <userName>username</userName>
  <agentId>8007</agentId>
  <canRemove>true</canRemove>
</agent>
```

When doing create or update, only the refURL parameter is required. Additional parameters are ignored. For example:

```
<agent>
   <team>
     <refURL>/unifiedconfig/team/5000</refURL>
   </team>
  …
</agent>
```

Items can also contain a collection of references. For example, if an agent belongs to multiple skill groups, the skillGroups parameter contains a reference to each associated skill group:

```
<agent>
  <skillGroups>
    <skillGroup>
      <refURL>/unifiedconfig/config/skillgroup/5001</refURL>
      <name>FirstSkill</name>
    </skillGroup>
    <skillGroup>
      <refURL>/unifiedconfig/config/skillgroup/5005</refURL>
      <name>AnotherSkill</name>
    </skillGroup>
  </skillGroups>
  …
</agent>
```

If the referenced item belongs to a department, then department information is included within that reference:

```
<agentDeskSettings>
   <refURL>/unifiedconfig/config/agentdesksetting/5001</refURL>
   <name>mike</name>
   <department>
      <refURL>/unifiedconfig/config/department/5003</refURL>
      <name>d1</name>
   </department>
</agentDeskSettings>
```

If the referenced item is a global item, then department information is omitted from the reference.

XML

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.

Error Responses

Operations that fail return an HTTP status code (HTTP 1.1 Status Codes) indicating if there was a client error or server error. The body of the response contains a collection of API error items to provide additional information about the failure.

Parameters

- errorType: Indicates the type of error. This is the primary identifier for the problem and can be used to map the type to a user readable string. For example, if your application receives an error with the errorType of invalidInput.fieldRequired, then you could display "This field is a required field; it cannot be left blank" to the user.
- errorData: The name of the parameter that had the error.
- errorMessage: Extra information about the error that is intended for the developer. This information is typically a sentence or other string. It is not localized, so it should not be shown to the user.
- errorDetail: Some errors contain additional detail parameters that are included in the errorDetail parameter.
    - If the error type is invalidInput.outOfRange, then errorDetail includes the following parameters:
        - min: The minimum value allowed.
        - max: The maximum value allowed.
    - If you attempt to delete an item that is in use by other items, the errorType is referenceViolation.api and the errorDetail includes the following parameters:

- referenceType: The type of item that references the item you tried to delete.
- references: A collection of references, referencing the item you tried to delete, including the name and refURL of each referencing item.
- totalCount: The total number of items referencing the item you attempted to delete.
- totalShown: The total number of items included in the references collection.

Example Error Response

The following error is returned when attempting to create a call type with a negative value for the serviceLevelThreshold parameter:

```
<apiErrors>
    <apiError>
        <errorData>serviceLevelThreshold</errorData>
        <errorDetail>
            <min>1</min>
            <max>2147483647</max>
        </errorDetail>
        <errorMessage>This field must contain a value from 1 to 2147483647</errorMessage>
        <errorType>invalidInput.outOfRange</errorType>
    </apiError>
</apiErrors>
```

Pagination

Pagination allows you to limit the number of items returned by the list operation and provides information on how to get other pages.

Query Parameters

- startIndex: Specfies the index of the item at which to start. Zero-based: 0 is the first item.
- resultsPerPage: Specifies the number of items to retrieve. Minimum:1. Default: 25. Maximum: 100.

Returned Parameters

- totalResults: Total number of items.
- resultsPerPage: Number of items requested per page.
- startIndex: The index of the first item returned. If you request a startIndex that is greater than total items, a full last page is returned.
- nextPage: The URL to get the next page. This parameter is not returned if you are on the last page.
- prevPage: The URL to get the previous page. This parameter is not returned if you are on the first page.
- firstPage: The URL to get the first page.
- lastPage: The URL to get the last page.
- searchTerm: The value specified in the search query parameter. See Search, on page 10.
- sortTerm: The value specified in the sort query parameters. See Sort, on page 11.

Note Query parameters for search and sort are included in the URL.

Example Response

```
<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>
```

Shared Parameters

changeStamp

- The version of the item. Initially set during a create (create, on page 3) operation.
- A changeStamp is a required parameter for the body of a PUT (update, on page 4) operation for items. 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 changeStamp is incremented.

description

- A description for this item.
- Optional parameter.
- No restriction of characters; OEM locale supported characters are allowed. For information on how to configure your system to support native character sets, see the latest version of the document *Cisco Packaged Contact Center Enterprise Installation and Upgrade Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-installation-guides-list.html.
- Maximum length of 255 characters.

name

- Required parameter.
- Maximum length of 32 characters allowed.
- Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric.
- Does not allow internationalized characters.

refURL

- The identifier for an item.
- Read-only parameter.

Permissions

Permissions information is included in list responses to indicate the write operations that the user is allowed to perform. If the API does not support any write operations, then permissions information is not returned.

Parameters

- canCreate: Indicates whether a create operation is allowed. Values are true/false. If the create operation is not supported by the API, then this parameter is not returned.
- canUpdate: Indicates whether an update operation is allowed. Values are true/false. If the update operation is not supported by the API, then this parameter is not returned.
- canDelete: Indicates whether a delete operation is allowed. Values are true/false. If the delete operation is not supported by the API, then this parameter is not returned.
- role: Type of role of the user performing the request. Values are administrator/supervisor.
- departmentAdmin: Indicates whether or not an administrator is restricted to specific departments only (true), or is a global administrator (false). Values are true/false.

Example Get Response

```
<permissionInfo>
    <canCreate>false</canCreate>
    <canUpdate>true</canUpdate>
    <canDelete>false</canDelete>
    <role>Administrator</role>
    <departmentAdmin>false</departmentAdmin>
</permissionInfo>
```

Synchronous vs. Asynchronous Writes

Synchronous API calls are blocking calls that do not return until either the change has been completed or there has been an error. For asynchronous calls, 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 the status than to wait for each call to complete before submitting the next one.

The following examples describe how to use the asynchronous feature to create a call type.

Performing Asynchronous Operations

The create, update, and delete operations can be performed asynchronously by including the query parameter `async=true`. The request is accepted if the operation is valid and the number of outstanding requests does not exceed the capacity. If the request is accepted, the response includes the following items:

- The response code is HTTP 202, indicating that the request has been accepted for processing.

- The location header specifies a URL that can be polled to receive updated information on the progress of the request.
- The response includes a body. See the next section **Asynchronous result parameters**.

Asynchronous Result Parameters

- progress: Indicates the current state of the request. Values include the following states:
  - IN_QUEUE: The request passed validation and capacity checks and was put in the queue.
  - IN_PROGRESS: The request is being processed.

Polling the Asynchronous Request Status

Use the URL from the location header of an asynchronous operation request to get updated status. Responses of this request are:

- If the request has not completed yet, the response contains the HTTP 202 response code, a location header with polling URL, and a response body.
- If the request has completed, the response is identical to the responses of synchronous operations, including the following:
  - For a successful create, the response code is HTTP 201 and the location header has the URL of the created item.
  - For a successful update or delete, the response code will be HTTP 200.
  - For an unsuccessful update, a body will provide information about the failure.
- If the request has been in queue for over 30 seconds, then it is removed and an error indicates that the request timed out.

Search

The list operation can be modified to return data you are looking for by applying the search query parameter.

Default Search Parameters

Typically, the name and description fields are searched when specifying a search string. Refer to each API section for the default search parameters permitted. For example, a query parameter of `q=abc` causes the list operation to return only entries with a name or description containing **abc**. The search value for default parameters has the following behaviors and restrictions:

Values:

- Are case-insensitive.
- Can be contained anywhere in the parameter value.
- Can match any of the default parameters.
- Cannot include SQL wildcards. They are not supported.
- Must be URL encoded. For example, **&** must be converted to **%26** so that it is not treated as a separator for additional query parameters.

Advanced Search

Advanced search parameters allow specific parameters to be searched. Refer to each API section for the advanced search parameters permitted. Advanced search parameters can be combined with a default search value. For example, applying the search query parameter of `q=abc routingType:1` to a dialed number list operation returns results where the routingType is set to one, and one of the default search parameters contains **abc**. Advanced search also has the following restrictions:

- Search terms must be separated by a space.
- Search terms can be specified in any order.

Advanced Departmental Search

For APIs that include a department reference, the following advanced search parameters are allowed:

- **departments:** Search for items that belong to the any of the specified departments. For example: `q=departments:(dept1|dept2)` search for items that belong to a department whose name exactly matches **dept1** or **dept2**. The departments search must follow these guidelines:
  - Department names must match exactly.
  - Departments are separated by the pipe character.
  - Up to 10 departments can be specified.
- **globaldepartment:** Specifies if the search should return items from the global department. Values include:
  - both: Return items from the global department and those specified in the departments search.
  - only: Return items from the global department only. The departments search term is ignored.
  - none: Do not return items from the global department.

Sort

A sort query parameter can be used to specify the order of the results in a list response.

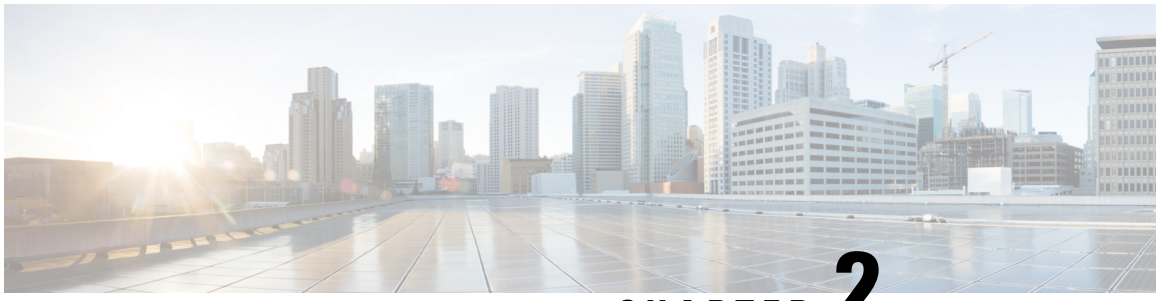
The query parameter is **sort=<parameterName>** order, where:

- parameterName: The name of the parameter that you want to sort on. This is case sensitive, so it must match the parameter in the API exactly.
- order: Specifies the order of the sort. Values are as follows:
  - asc: Perform an ascending sort. This is the default if no order is specified.
  - desc: Perform a descending sort.

Example

For example, to find all the CallTypes whose name or description contains *supervisor*, sorted in ascending order by *name*:

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



CHAPTER 2

Active Directory Domain API

Use the Active Directory Domain API to list the active directory domains currently defined in your call center environment. It is read-only, and does not require authentication.

URL

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

Operations

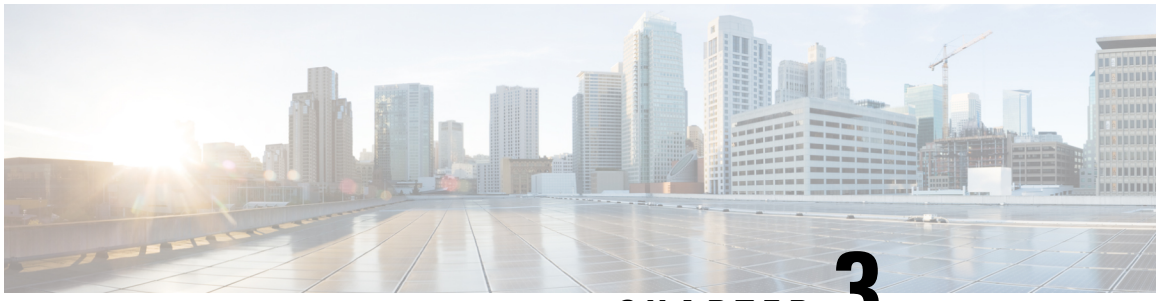
- list: Retrieves a list of active directory domains.

Parameters

- name: The name of the domain.
- systemDomain: Indicates if the system is a member of this domain. Values are true/false.

Example List Response

```
<results>
  <activeDirectoryDomains>
    <activeDirectoryDomain>
       <name>boston.com</name>
       <systemDomain>true</systemDomain>
    </activeDirectoryDomain>
    <activeDirectoryDomain>
       <name>cisco.com</name>
       <systemDomain>false</systemDomain>
    </activeDirectoryDomain>
  </activeDirectoryDomains>
</results>
```



CHAPTER 3

Administrator API

An administrator is an Active Directory user who has been provided access to the system.

That access can be controlled by assigning the administrator to roles and departments (see Role API, on page 187 and Department API, on page 89) .

Use the Administrator API to list the administrators currently defined in the database, define new administrators, and view, edit, and delete existing administrators.

URL

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

Operations

- create: Creates one administrator.
- delete: Permanently deletes one administrator.
- get: Returns one administrator, using the URL `https://<server>/unifiedconfig/config/administrator/<id>`.
- list: Retrieves a list of administrators.
- update: Updates one administrator.

Parameters

- refURL: The refURL of the administrator. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- userName: Required. The unique username of an existing Active Directory account. Maximum length of 64 characters.
- domainName: The domain for this administrator. If blank, system uses the default domain name. Maximum length of 64 characters.
- departments: A collection of department (Department API, on page 89) references associated with this administrator, including the refURL and name. Leave this collection empty to allow the administrator to have access to all departments. See References, on page 5.

- role: A reference to a role (Role API, on page 187), including refURL and name. This parameter sets access to specific features. Automatically creates membership to Active Directory Setup group or Config group. If no role is assigned, the administrator is not placed in either group and does not have access to any of the web configuration tools, the Configuration Manager, or the Script Editor. See References, on page 5.
- readOnly: Required. Specifies whether the administrator has read-only access to the APIs and tools. Values are true/false.

Search and Sort Values

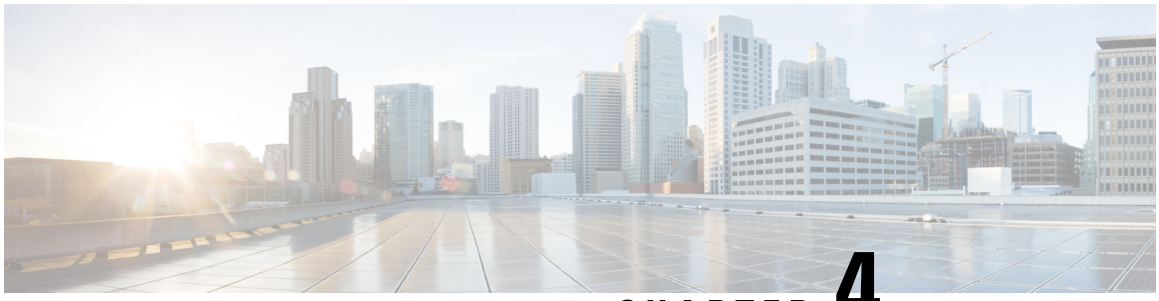
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• userName • domainName • description	• userName • domainName • description

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<administrator>
      <changeStamp>3</changeStamp>
      <refURL>/unifiedconfig/config/administrator/5000</refURL>
      <domainName>domain</domainName>
      <userName>user1</userName>
      <departments>
        <department>
          <refURL>/unifiedconfig/config/department/5000</refURL>
          <name>dept1</name>
        </department>
        <department>
          <refURL>/unifiedconfig/config/department/5001</refURL>
          <name>dept2</name>
        </department>
      </departments>
      <description>desc</description>
      <readOnly>true</readOnly>
      <role>
        <refURL>/unifiedconfig/config/role/5005</refURL>
        <name>ConfigAdmin</name>
      </role>
</administrator>
```



CHAPTER 4

Agent API

Agents respond to contacts from customers. Use the Agent API to list the agents currently defined in the database, define new agents, and view, edit, and delete existing agents.

Note In the Role API, when you enable the ManageAgentAttribute or ReSkillAgents subfeature in the accessList parameter for a custom role then the Agent API is provided with Update Only Access instead of Full Access. With Update Only Access, you cannot create and delete an agent using Agent API.

URL

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

Operations

- create: Creates an agent.
- delete: Marks one agent for deletion, but does not permanently delete the agent.

  Note When you delete a supervisor's record in Packaged CCE, the corresponding user account in Unified Intelligence Center is also deleted.

  When you delete an agent record, its association with the teams is removed and the same is updated in the corresponding collection in Unified Intelligence Center.

- get: Returns one agent, using the URL `https://<server>/unifiedconfig/config/agent/<id>`.
- list: Retrieves a list of agents.
  - **Query parameters:**
    - selectedAttribute: Use this query parameter to augment the returned agent parameters with an extra parameter called selectedAttribute. This parameter indicates if the agent belongs to the attribute with the ID specified in this query parameter. For example, to find out which agents belong to the specified attribute, add `selectedAttribute=5000`.

Note Using selectedAttribute automatically sets the summary list query parameter to true.

- selectedSkillGroup: Use this query parameter to augment the returned agent parameters with an extra parameter called selectedSkillGroup. This parameter indicates if the agent belongs to the skill group with the ID specified in this query parameter. For example, to find out which agents belong to the specified skill group, add `selectedSkillGroup=5001`.
- ignoreRole: Use this query parameter to allow a supervisor to see a list of all agents in the system, including agents the supervisor does not supervise. For example, to see all agents, add `ignoreRole=true`.
- Summary list: See list, on page 3.

- update: Updates one agent.

Note
- When you change the team association for an agent or supervisor in Packaged CCE, the same change is updated in the corresponding collection in Unified Intelligence Center.
- When you change the username and team association for a supervisor's record, the same changes are also updated in the corresponding user account in Unified Intelligence Center.
- For an existing supervisor's record in Packaged CCE, if the value for the Supervisor parameter is set to false, the corresponding user account is deleted from Unified Intelligence Center.

Parameters

- refURL: The refURL for the agent. See Shared Parameters, on page 8.
- agentId: The unique peripheral number. Maximum length of 11 characters allowed. Default is an autogenerated 7-digit number.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- department: A reference to the agent's department (Department API, on page 89), including the refURL and name. See References, on page 5.
- agentStateTrace: Indicates if agent state tracing is turned on for the agent. True or false. See Agent State Trace API, on page 29.
- agentDeskSettings: A reference to the agent's agentDeskSettings (Agent Desk Settings API, on page 25), including the refURL and name. See References, on page 5.
- person: Required. Includes the following parameters:

- firstName: Agent's first name. Maximum of 32 characters. International characters are allowed.
- lastName: Agent's last name. Maximum of 32 characters. International characters are allowed.
- password: Agent's password. Maximum of 256 ASCII characters. Password is case-sensitive. The password can be used when creating or updating, but is not returned.

  If the ssoEnabled parameter is set to true, the password is not saved.
- userName: Agent's login name. Maximum of 255 ASCII characters. Must be unique. The login name supports the use of all characters from 33 to 126 in the ASCII character set, except for the following: double quotation mark ("), forward slash (/), backward slash (\), square brackets ([ ]), colon (:), semicolon (;), pipe (|), equal to (=), comma (,), plus sign (+), asterisk (*), question mark (?), angle brackets (< >), hash (#), percent (%), and SPACE.

  For supervisors and for agents with single sign-on (SSO) enabled, the username is the user's Active Directory or SSO account username.

  For supervisors who are not enabled for single sign-on (SSO), the Active Directory username must be in the user@domain format.
- loginEnabled: Whether the agent can log in. True or false. Default is true.
- ssoEnabled: Whether single sign-on is supported at the agent level. True or false. Default is false. This parameter takes effect only when the global level SSOEnabled is mixed.

- ecePerson: If the agent is ECE-enabled or not. True or false. Default is false.
  - screenName: The screen name of the agent. Default is null. Maximum characters is 30.

    Supports all characters, except for the following:

    Period (.), hash (#), asterisk (*), less than (<), greater than (>), comma (,), question mark (?), colon (:), dollar ($), forward slash, (/), and backward slash (\).
  - emailAddress: The email address of the agent. Default is null. Maximum characters is 50.

Note The screenName and emailAddress parameters are applicable only for the ECE agent.

- supervisor: Required. Indicates whether the agent is marked as supervisor. True or false.

  If set to true, the person userName is used for the supervisor username and domain.

Note

- When the value for this parameter is set to true, a user account is created in Unified Intelligence Center with the supervisor's username and domain name. If the username and domain name of the supervisor already exists in Unified Intelligence Center, the Unified Intelligence Center user account and supervisor's record are synchronized to have same username and domain name.
- When you add teams to a supervisor's record, the same teams are also added to the corresponding user account (with Supervisor permission) in Unified Intelligence Center.

- agentAttributes: A collection of agent attribute (Attribute API, on page 43) references for this agent, including the refURL, and read-only parameters name, dataType, and description for each associated attribute. Also includes the attributeValue parameter which indicates the value (true or false or 1-10), and description of the attribute for this agent. See References, on page 5.
- agentAttributesAdded: A collection of agent attribute references (Attribute API, on page 43) to be added to the agent, including the agent refURL and the attributeValue of each attribute. If the attributeValue is not specified, it is assigned the default value. Agents that already have this attribute are updated with the specified attributeValue. This parameter is update only, and cannot be used with the agentAttributes parameter. This parameter can be used with the agentAttributesRemoved parameter. See References, on page 5.
- agentAttributesRemoved: A collection of agent attribute references (Attribute API, on page 43) to be removed from the agent, including the refURL of each attribute. This parameter is update only, and cannot be used with the agentAttributes parameter. This parameter can be used with the agentAttributesAdded parameter. See References, on page 5.
- selectedAttribute: Indicates if the agent has the specified attribute. Returned only when using the selectedAttribute query parameter. True or false.
- skillGroups: A collection of skill group references for this agent (Skill Group API, on page 221), including the refURL and name of each associated skill group. See References, on page 5.
- skillGroupsAdded: A collection of skill group references to be added to the agent, including the refURL of each skill group to be added. This parameter is update only, and cannot be used with the skillGroups parameter. This parameter can be used with the skillGroupsRemoved parameter. See References, on page 5.
- skillGroupsRemoved: A collection of skill group references to be removed from the agent, including the refURL of each skill group to be removed. This parameter is update only, and cannot be used with the skillGroups parameter. This parameter can be used with the skillGroupsAdded parameter. See References, on page 5.
- defaultSkillGroup: A reference to a skill group, including the refURL and name. Identifies the default skill group associated with this agent. See References, on page 5.
- selectedSkillGroup: Indicates if the agent has the specified skill group. Returned only when using the selectedSkillGroup query parameter. True or false.

Note Using selectedSkillGroup automatically sets the summary list query parameter to true.

- agentTeam: A reference to the agent's team (Agent Team API, on page 31), including the refURL and name. See References, on page 5.
- supervisorTeams: If this agent has supervisor access, this collection of references is for this supervisor's teams, including the refURL and name of each supervised team. See References, on page 5.
- peripheralSet: This parameter is mandatory for Packaged CCE 4000 Agents or 12000 Agents deployment type. You must provide the reference to a peripheral set for which Agent PG is configured.

  The perpheralSet parameter is not available for Packaged CCE 2000 Agents deployment type.
- datacenter: A reference to the data center, including the refURL and name.

  You must provide the reference to a data center that contains above peripheral set. For more information on data center for 4000 Agents or 12000 Agents deployment, see Inventory Import API, on page 123.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• agentId • description • person.firstName • person.lastName • person.userName	• agentId • description • supervisor • agentStateTrace • agentTeam.name • person.firstName • person.lastName • person.userName • person.loginEnabled • person.ssoEnabled • datacenter.name • peripheralSet.name This parameter is available for Packaged CCE 4000 Agents and 12000 Agents deployment type.

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

There are several advanced searches you can perform on the Agent API, including supervisor, attributes, skillgroups, team, data centers, and include and exclude (agentId).

- **supervisor: (true/false)** Find agents that are (or are *not*) supervisors.
  - **q=supervisor:true** Returns all agents who are supervisors.
  - **q=supervisor:false** Returns all agents who are *not* supervisors.
- **eceagent: (true/false)** Find agents that are (or are *not*) Enterprise Chat and Email (ECE) agents.
  - **q=eceagent:true** Returns all ECE agents.
  - **q=eceagent:false** Returns all agents who are *not* ECE agents.
- **attributes: (attr1 & attrt2 & attr3, ...)** find *all* agents that have *all* the specified attributes. Up to ten attributes can be specified. The attribute names are fully matched.
- **skillgroups: (skill1 & skill2 & skill3,...)** find *all* agents that have *all* the specified skillgroups. Up to ten skillgroups can be specified. The skillgroup names are fully matched.
- **team: (team1|team2|team3, ...)** find *all* agents who belong to *any* of the specified teams. Up to ten team names can be specified. The team name is fully matched.
- **datacenters: (dc1|dc2|dc3...)** which returns all agents who belong to any of the specified data centers. Up to three data centers can be specified. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all machines in the core data center.
- **include: (ID1 & ID2 & ID3, ...)** find *all* specified agents even if they do not meet other search criteria. Each ID is fully matched. Obtain this ID from the refURL. For example, 5017 is the ID in the following refURL `<refURL>/unifiedconfig/config/agent/5017</refURL>`.
- **exclude: (ID1 & ID2 & ID3, ...)** exclude *all* specified agents from the results even if they meet all other search criteria. Each ID is fully matched. Obtain this ID from the refURL. For example, 5017 is the ID in the following refURL `<refURL>/unifiedconfig/config/agent/5017</refURL>`.
- **peripheralsets: (ps1|ps2|ps3...)** returns the agents specific to the peripheral sets in 4000 Agents/12000 Agents deployment. The peripheral set names are fully matched (case-insensitive, no partial matches).

Example Get 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@xyz.com</userName>
             <password>mypassword</password>
             <ssoEnabled>false</ssoEnabled>
           <ecePerson>true</ecePerson>
             <emailAddress>agent@xyz.com</emailAddress>
             <screenName>agentScreenName</screenName>
       </person>
```

```
        <agentDeskSettings>
             <name>test2</name>
             <refURL>/unifiedconfig/config/agentdesksetting/5434</refURL>
        </agentDeskSettings>
        <datacenter>
                <name>Berlin</name>
                <refURL>/unifiedconfig/config/datacenter/5000</refURL>
        </datacenter>
        <supervisor>true</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>
         <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>
```

Example Get Response for Packaged CCE 4000 Agents and 12000 Agents Deployment

```
<agent>

  <refURL>/unifiedconfig/config/agent/5004</refURL>
  <changeStamp>0</changeStamp>
  <agentId>7132334</agentId>
  <agentStateTrace>false</agentStateTrace>
  <datacenter>
  <refURL>/unifiedconfig/config/inventory/datacenter/<datacenter-id></refURL>
    <name>core</name>
  </datacenter>
  <peripheralSet>
   <refURL>/unifiedconfig/config/inventory/datacenter/core/peripheralset/5001</refURL>
```

```
   <name>ps1</name>
  </peripheralSet>
  <person>
   <firstName>agent3</firstName>
   <lastName>last</lastName>
   <loginEnabled>true</loginEnabled>
   <ssoEnabled>false</ssoEnabled>
   <userName>agent3</userName>
   <ecePerson>true</ecePerson>
      <emailAddress>agent@xyz.com</emailAddress>
      <screenName>agentScreenName</screenName>
 </person>
 .....
 ......
 <supervisor>false</supervisor>
</agent>
```

REST Responses

Following are the possible REST responses that can be received for Agent API calls:

- **Success (201 Created or 200 OK)**

  Configuration changes persist in AW DB and synchronized with ECE and CUIC.

- **Partial Success (201 Created or 200 OK)**

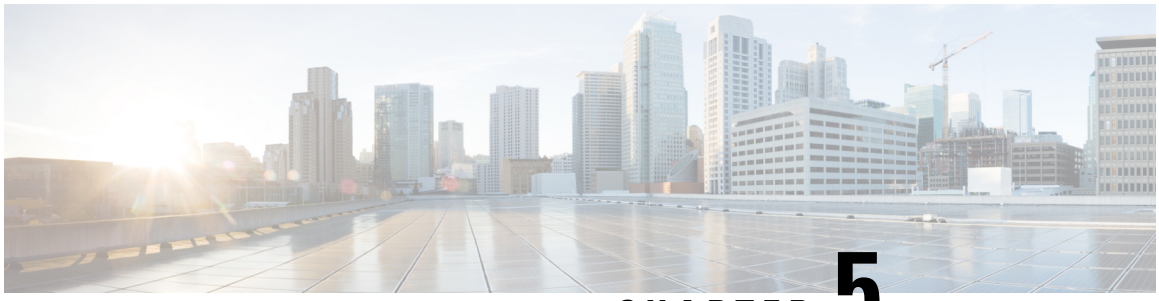
  Configuration changes are persist in AW DB, but failed to synchronize with ECE and CUIC. In this case even if the REST response status is a success (201 Created or 200 OK), the response body will include an APIError.

  Examples of API errors:

  ```
  <apiErrors>
   <apiError>
    <errorMessage>Configuration update failed for one or more devices.</errorMessage>
    <errorType>PARTIAL_SUCCESS</errorType>
   </apiError>
  </apiErrors>
  ```

- **Server Busy (503 Service Unavailable)**

  This occurs when data synchronization to a device is in progress.



CHAPTER 5

Agent Desk Settings API

A desk settings is a collection of permissions or characteristics for the agent, such as how and when calls to the agent are redirected, how and when the agent enters various work states, and how requests to the supervisor are handled.

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, and delete existing agent desk settings.

URL

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

Operations

- create: Creates one agent desk settings.
- delete: Permanently deletes one agent desk settings.
- get: Returns one agent desk settings, using the URL `https://<server>/unifiedconfig/config/agentdesksetting/<id>`.
- list: Retrieves a list of agent desk settings.
- update: Updates one agent desk settings.

Parameters

- refURL: The refURL of the agent desk settings. See Shared Parameters, on page 8.
- name: The name of the agent desk settings. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- department: A reference to the department (Department API, on page 89), including refURL and name. See References, on page 5.
- wrapupDataIncomingMode: Indicates whether the agent is allowed or required to enter wrap-up data after an inbound call.
    - 0: Required

    - 1: Optional (Default)
    - 2: Not Allowed

- wrapupDataOutgoingMode: Indicates whether the agent is allowed or required to enter wrap-up data after an outbound call.
    - 0: Required
    - 1: Optional (Default)
    - 2: Not Allowed

- remoteAgentType: Indicates if agents are allowed to login as remote agents.
    - 0: Not Allowed
    - 1: Call by Call
    - 2: Nailed Up
    - 3: Agent Chooses

- logoutNonActivityTime: Number of seconds of non-activity at the desktop after which the software automatically logs out the agent. Value must be between 10 and 7200 seconds (default is NULL).
- workModeTimer: Specifies the auto wrap-up time out. Value must be between 1 and 7200 seconds (default is 7200).
- supervisorAssistCallMethod: Indicates how the supervisor assist request call is made.
    - 0: Consultative Call (Default)
    - 1: Blind Conference

- emergencyCallMethod: Indicates how the emergency call request is made.
    - 0: Consultative Call (Default)
    - 1: Blind Conference

- idleReasonRequired: Indicates whether the agent must enter a reason before entering the Idle state. Values are true/false.
- logoutReasonRequired: Indicates whether or not the agent must enter a reason before logging out. Values are true/false.
- autoAnswerEnabled: Indicates whether or not calls sent to this agent will be answered automatically. Values are true/false.
- agentStateAfterRONA: Indicates the agent state after RONA. Values are notReady or ready.

Search and Sort Values

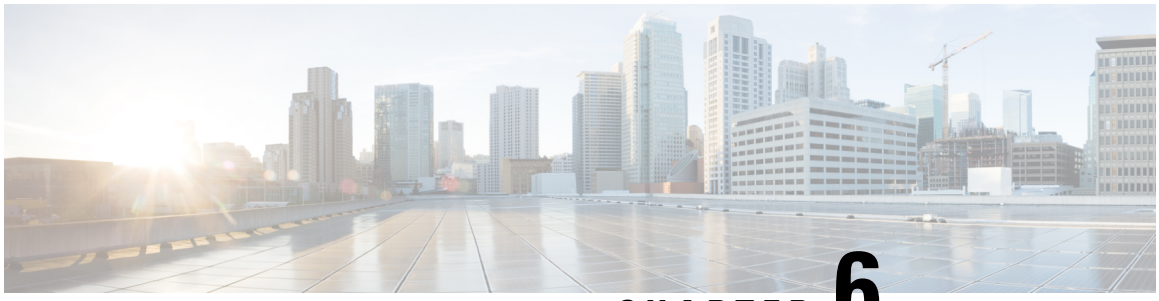
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description • wrapupDataIncomingMode • wrapupDataOutgoingMode • remoteAgentType • logoutNonActivityTime • workModeTimer • supervisorAssistCallMethod • emergencyCallMethod • idleReasonRequired • logoutReasonRequired • autoAnswerEnabled

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<agentDeskSetting>
      <changeStamp>3</changeStamp>
      <refURL>/unifiedconfig/config/agentdesksetting/5000</refURL>
      <agentStateAfterRONA>notReady</agentStateAfterRONA>
      <autoAnswerEnabled>false</autoAnswerEnabled>
      <emergencyCallMethod>0</emergencyCallMethod>
      <idleReasonRequired>false</idleReasonRequired>
      <logoutReasonRequired>false</logoutReasonRequired>
      <name>Default_Agent_Desk_Settings</name>
      <remoteAgentType>0</remoteAgentType>
      <supervisorAssistCallMethod>0</supervisorAssistCallMethod>
      <workModeTimer>7200</workModeTimer>
      <wrapupDataIncomingMode>1</wrapupDataIncomingMode>
      <wrapupDataOutgoingMode>1</wrapupDataOutgoingMode>
</agentDeskSetting>
```



CHAPTER 6

Agent State Trace API

Enabling agent trace allows you to track and report on every state an agent passes through. Use this feature for short-term tracking of specific agents.

Note The maximum number of agents with AgentStateTrace on is 100.

URL

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

Operations

- list: Returns a list of agents whose agent state trace is turned on.
- update: Updates the agent state trace in the database.

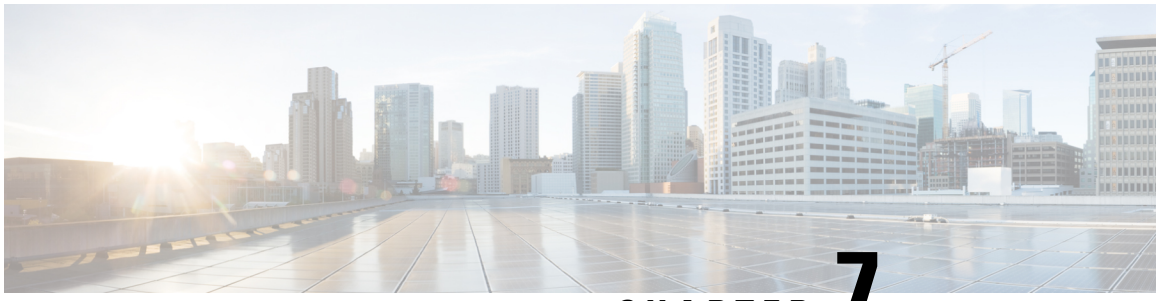
Parameters

- refURL: The refURL for agent state trace. See Shared Parameters, on page 8.
- agents: A collection of agent references (Agent API, on page 17). Each reference contains person (including firstName, lastName, userName, and loginEnabled parameters), agent refURL, agentId, supervisor, and agentStateTrace. Agents who are not specified in this collection have agentStateTrace turned off. To turn off all the agent state trace, pass in an empty list. See References, on page 5.

Example Get Response

```
<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>John</firstName>
                   <lastName>Doe</lastName>
                   <loginEnabled>true</loginEnabled>
```

```
                    <userName>jdoe</userName>
                  </person>
                <supervisor>false</supervisor>
              </agent>
        </agents>
</agentstatetrace>
```



CHAPTER 7

Agent Team API

You can associate a set of agents to a team with a specific supervisor. The supervisor can run reports on that team and receive Supervisor Assist requests from its members.

You can use the Agent Team API to list the agent teams currently defined in the database, define new agent teams, and view, edit, and delete existing agent teams.

URL

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

Operations

- create: Creates an agent team.

  Note
  - When you create a team in Packaged CCE, the same team record is also created as a collection in Cisco Unified Intelligence Center.
  - When you add Agents or Supervisors to a team, the same Agents or Supervisors (with Supervisor permission) are added to the corresponding collection in Unified Intelligence Center.

- delete: Permanently deletes one agent team from the database.

  Note When you delete a team in Packaged CCE, the corresponding collection is also deleted in Unified Intelligence Center.

- get: Returns one agent team, using the URL `https://<server>/unifiedconfig/config/agentteam/<id>`.
- list: Retrieves a list of agent teams.
  - **Query parameters:**
    - Summary list: See list, on page 3.

- update: Updates one agent team.

Note When you update an existing team record (For example, you add or remove the Agents or Supervisors) in Packaged CCE, the same changes are updated in the corresponding collection in Unified Intelligence Center.

Parameters

- refURL: The refURL of the agent team. See Shared Parameters, on page 8.
- name: The name of the agent team. See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- department: A reference to the department (Department API, on page 89), including the name and refURL. See References, on page 5.
- dialedNumber: A reference to an internal dialed number (Dialed Number API, on page 95) for the agent team, including the refURL and dialed number string. See References, on page 5.
- agents: A collection of agent (Agent API, on page 17) references, including the refURL, first name, last name, username, and agent ID for each agent on the team. See References, on page 5.
- agentCount: Read-only field. Number of agents on the team.
- supervisors: A collection of supervisor (Agent API, on page 17) references, including the refURL, first name, last name, username, and agent ID for each supervisor who supervises this team. See References, on page 5.
- supervisorCount: Read-only field. Number of supervisors who supervise this team.
- peripheralSet: This parameter is mandatory for Packaged CCE 4000 Agents or 12000 Agents deployment type. You must provide the reference to a peripheral set for which Agent PG is configured.

  The perpheralSet parameter is not available for Packaged CCE 2000 Agents deployment type.
- datacenter: The data center to which the agents on this team belong, including the refURL and name.

  You must provide the reference to a data center that contains above peripheral set. For more information on data center for 4000 Agents or 12000 Agents deployment, see Inventory Import API, on page 123.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description • datacenter.name • peripheralSet.name (Available for Packaged CCE 4000 Agents and 12000 Agents deployment types)

See Search, on page 10 and Sort, on page 11.

Advanced Search Parameters

datacenters: (dc1|dc2|dc3...) which returns all teams who belong to any of the specified data centers. You can specify up to three data centers. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all machines in the core data center.

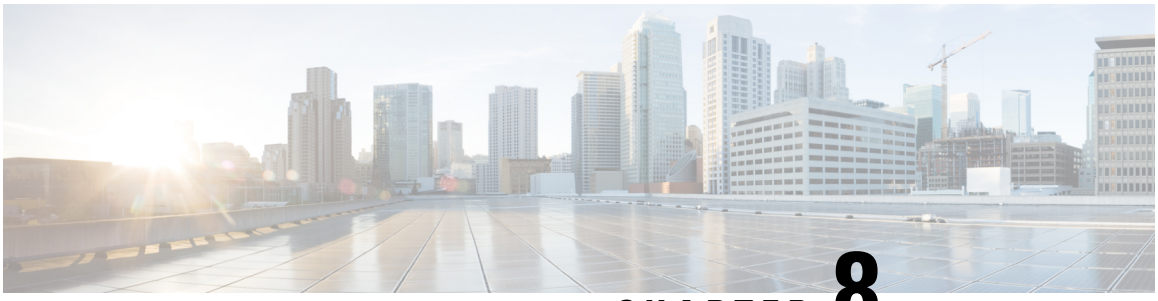
peripheralSets:(ps1|ps2|ps3...) returns the teams specific to the peripheral sets in 4000 Agents/12000 Agents deployment. The peripheral set names are fully matched (case-insensitive, no partial matches) .

Example Get Response

```
<agentTeam>

<refURL>https://***.***.***.***/unifiedconfig/config/agentteam/(id)</refURL>
    <name>team1</name>
    <datacenter>
        <refURL>/unifiedconfig/config/datacenter/9887</refURL>
        <name>Boston</name>
    </datacenter>
    <dialedNumber>
     <refURL>[https://***.***.***.***/unifiedconfig/config/dialednumber/(id)]</refURL>
       <dialedNumberString>8885551212</dialedNumberString>
    </dialedNumber>
    <description>test agent team1</description>
    <agentCount>1</agentCount>
    <agents>
       <agent>
          <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_1)]</refURL>
          <firstName>John</firstName>
          <lastName>Smith</lastName>
          <userName>username</userName>
          <agentId>8006</agentId>
       </agent>
       <agent>
          <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_2)]</refURL>
          <firstName>Jane</firstName>
          <lastName>Doe</lastName>
          <userName>username</userName>
          <agentId>8007</agentId>
       </agent>
    </agents>
    <supervisorCount>2</supervisorCount> <supervisor>
    <supervisors>
       <supervisor>
          <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_3)]</refURL>
          <firstName>Mary</firstName>
          <lastName>Hart</lastName>
          <userName>username</userName>
          <agentId>8008</agentId>
       </supervisor>
       <supervisor>
          <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_4)]</refURL>
          <firstName>Jack</firstName>
          <lastName>Jones</lastName>
          <userName>username</userName>
          <agentId>8009</agentId>
       </supervisor>
    </supervisors>
    <changeStamp>0</changeStamp>
 </agentTeam>
```

Example Get Response for Packaged CCE 4000 Agents or 12000 Agents Deployment Type

```
<agentTeam>

 <refURL>https://***.***.***.***/unifiedconfig/config/agentteam/(id)</refURL>
 <name>team1</name>
 <datacenter>
    <refURL>/unifiedconfig/config/datacenter/9887</refURL>
 <name>Boston</name>
 </datacenter>
 <dialedNumber>
 <refURL>[https://***.***.***.***/unifiedconfig/config/dialednumber/(id)]</refURL>
 <dialedNumberString>8885551212</dialedNumberString>
 </dialedNumber>
 <description>test agent team1</description>
 <agentCount>1</agentCount>
 <agents>
 <agent>
 <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_1)]</refURL>
 <firstName>John</firstName>
 <lastName>Smith</lastName>
 <userName>username</userName>
 <agentId>8006</agentId>
 </agent>
 <agent>
 <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_2)]</refURL>
 <firstName>Jane</firstName>
 <lastName>Doe</lastName>
 <userName>username</userName>
 <agentId>8007</agentId>
 </agent>
 </agents>
 <supervisorCount>2</supervisorCount>
 <supervisors>
 <supervisor>
 <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_3)]</refURL>
 <firstName>Mary</firstName>
 <lastName>Hart</lastName>
 <userName>username</userName>
 <agentId>8008</agentId>
 </supervisor>
 </supervisors>
<peripheralSet>
<refURL>/unifiedconfig/config/inventory/datacenter/boston/peripheralset/5001</refURL>
<name>PS1</name>
</peripheralSet>
<changeStamp>0</changeStamp>
</agentTeam>
```



CHAPTER 8

Application Gateway API

Use this API to create a simple application gateway of the type *custom gateway*. You can have up to 20 application gateways.

URL

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

Operations

- create: Creates one application gateway and stores it in the database.
- delete: Deletes one application gateway from the database.
- get: Retrieves one application gateway from the database, using the URL `https://<server>/unifiedconfig/config/applicationgateway/<id>`
- list: Retrieves a list of application gateways, using the URL `https://<server>:<serverport>/unifiedconfig/config/applicationgateway/<id>`
- update: Updates one application gateway in the database, using the URL `https://<server>:<serverport>/unifiedconfig/config/applicationgateway/<id>`

Parameters

- refURL: The refURL for the application gateway. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- encryption: Indicates whether requests to the application gateway are encrypted. True or false. Default is false.
- faultTolerance: If the application gateway is duplexed, specifies the fault-tolerance strategy used. Possible values include the following:
  - NONE (default)
  - DUPLICATE_REQUEST
  - ALTERNATE_REQUEST

- HOT_STANDBY

- name: Required. An enterprise name for the application gateway.
- preferredSide: Required. The preferred side of the gateway to use when both sides are available. A or B.
- side: Indicates the side whether A or B.
- address: IP address or hostname to connect to.
- port: port number; to be used together with address field.
- initializationData: An optional string software passes to the host during initialization. This corresponds toconnectionInfofield.
- inService: Checks the connection availability.

Note If any of the following fields are present during create or update API, it overrides the global settings.

- maxErrors: Indicates the number of consecutive errors that cause the router to declare the host unavailable. This corresponds to *ErrorThreshold* column.
- heartBeat
  - interval: Indicates the number of milliseconds the router waits between successful heartbeats.
  - retry: Limit Number of consecutive unanswered heartbeats after which the router ends the connection. This corresponds to *HeartbeatLimit* column.
  - retryTimeout: Indicates the number of milliseconds the router waits before retrying a missed heartbeat. This corresponds to *HeartbeatRetry* column.
  - requestTimeout: Indicates the number of milliseconds the router waits for a response to a heartbeat before considering it as a failure. This corresponds to *HeartbeatTimeout* column.
- session:
  - openTimeout: Indicates the number of milliseconds the router waits for a response to an open or close request. If it receives no response within this time, the router assumes the request failed.
  - retryTimeout: Indicates the number of milliseconds the router waits before trying to reconnect after a connection terminates or a connection attempt fails. This corresponds to the *SessionRetry* column.
  - retryLimit: Indicates the number of times the router tries to establish the connection before it quits. This corresponds to the *SessionRetryLimit* column.
- timeout
  - abandon: An internal timeout in milliseconds to communicate between the router and the Application Gateway interface process.
  - late: Indicates the number of milliseconds the router waits for a response before considering it as late. This does not affect router processing. It is for statistical use only.
  - request: Indicates the number of milliseconds the system waits before timing out a request.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name • description • encryption

Note Only default search is supported.

See Search, on page 10 and Sort, on page 11.

Example Create Request

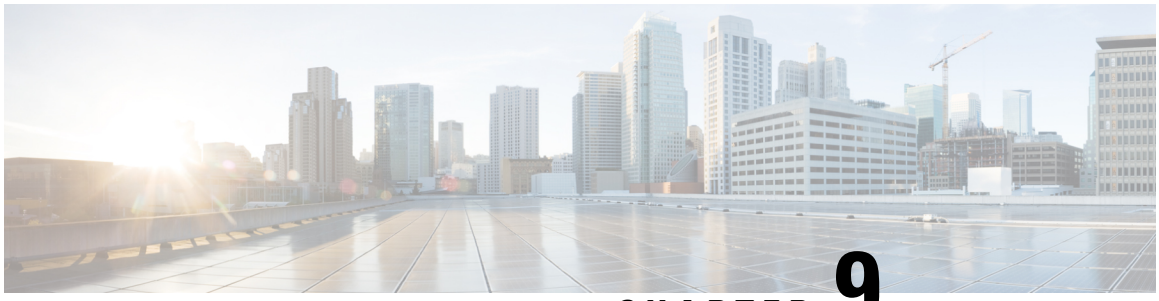
```
<applicationGateway>
    <description>application gateway for application1</description>
    <encryption>true</encryption>
    <faultTolerance>ALTERNATE_REQUEST</faultTolerance>
    <name>application gateway for application1</name>
    <preferredSide>A</preferredSide>
    <connections>
        <connection>
            <side>A</side>
            <address>120.22.30.102</address>
            <port>2033</port>
            <initializationData>connect</initializationData>
            <inService>true</inService>
            <maxErrors>10</maxErrors>
            <heartBeat>
                <interval>15000</interval>
                <retryLimit>10</retryLimit>
                <retryTimeout>200</retryTimeout>
                <requestTimeout>300</requestTimeout>
            </heartBeat>
            <session>
                <openTimeout>15000</openTimeout>
                <retryTimeout>60000</retryTimeout>
                <retryLimit>10</retryLimit>
            </session>
            <timeout>
                <abandon>5000</abandon>
                <late>300</late>
                <request>300</request>
            </timeout>
        </connection>
        <connection>
            <side>B</side>
            <address>120.22.30.104</address>
            <port>2034</port>
            <initializationData>connect</initializationData>
            <inService>true</inService>
            <maxErrors>10</maxErrors>
            <heartBeat>
                <interval>15000</interval>
```

```
                <retryLimit>10</retryLimit>
                <retryTimeout>200</retryTimeout>
                <requestTimeout>300</requestTimeout>
            </heartBeat>
            <session>
                <openTimeout>15000</openTimeout>
                <retryTimeout>60000</retryTimeout>
                <retryLimit>10</retryLimit>
            </session>
            <timeout>
                <abandon>5000</abandon>
                <late>300</late>
                <request>300</request>
            </timeout>
        <connection>
    </connections>
</applicationGateway>
```

Example Get Response

```
<applicationGateway>
    <refURL>/unifiedconfig/config/applicationgateway/5000</refURL>
    <changeStamp>1</changeStamp>
    <description>application gateway for application1</description>
    <encryption>true</encryption>
    <faultTolerance>ALTERNATE_REQUEST</faultTolerance>
    <name>appGateway1</name>
    <preferredSide>A</preferredSide>
    <connections>
        <connection>
            <side>A</side>
            <address>120.22.30.102</address>
            <port>2033</port>
            <initializationData>connect</initializationData>
            <inService>true</inService>
            <maxErrors>10</maxErrors>
            <heartBeat>
                <interval>15000</interval>
                <retryLimit>10</retryLimit>
                <retryTimeout>200</retryTimeout>
                <requestTimeout>300</requestTimeout>
            </heartBeat>
            <session>
                <openTimeout>15000</openTimeout>
                <retryTimeout>60000</retryTimeout>
                <retryLimit>10</retryLimit>
            </session>
            <timeout>
                <abandon>5000</abandon>
                <late>300</late>
                <request>300</request>
            </timeout>
        </connection>
        <connection>
            <side>B</side>
            <address>120.22.30.104</address>
            <port>2034</port>
            <initializationData>connect</initializationData>
            <inService>true</inService>
            <maxErrors>10</maxErrors>
            <heartBeat>
                <interval>15000</interval>
                <retryLimit>10</retryLimit>
```

```
                <retryTimeout>200</retryTimeout>
                <requestTimeout>300</requestTimeout>
            </heartBeat>
            <session>
                <openTimeout>15000</openTimeout>
                <retryTimeout>60000</retryTimeout>
                <retryLimit>10</retryLimit>
            </session>
            <timeout>
                <abandon>5000</abandon>
                <late>300</late>
                <request>300</request>
            </timeout>
        <connection>
    </connections>
</applicationGateway>
```



CHAPTER 9

Application Gateway Global Settings API

Use this API to retrieve global settings values for application gateway configurations. This is a read-only API.

URL

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

Operations

- list: Retrieves a list of values for the global configuration settings for each application gateway type. The only supported type is CUSTOM_GATEWAY.

Parameters

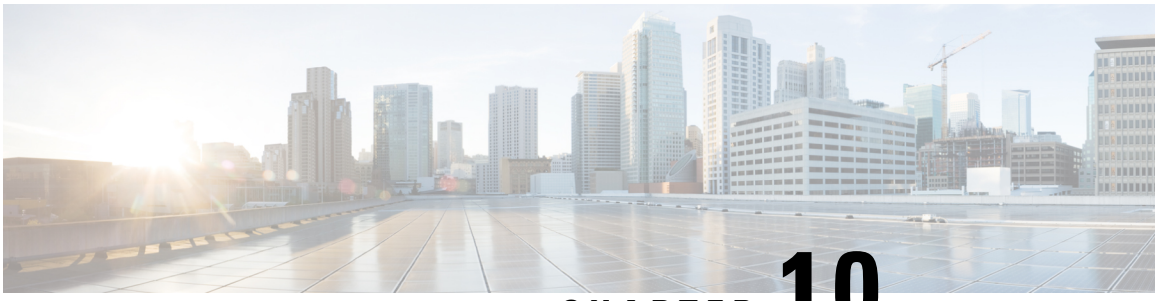
- errorThreshold: Specifies the number of consecutive errors after which the router declares the Application Gateway host unavailable and tries to reconnect.
- heartbeat: Information about the heartbeat, which includes the following parameters:
  - interval: The time in milliseconds the router waits between successful heartbeats.
  - limit: The number of consecutive missed heartbeats after which the router closes the connection and attempts to reconnect.
  - retry: How long in milliseconds the router waits before retrying a heartbeat after a heartbeat has failed.
  - timeout: How long in milliseconds the router waits for a response to a heartbeat before considering it a failure.
- session: Information about the session, which includes the following parameters:
  - openTimeout: How long in milliseconds the router waits for a response to an open or close request. If it receives no response within this time, the router assumes the request failed.
  - retry: How long in milliseconds the router waits before trying to reconnect after a connection terminates or a connection attempts fails.
  - retryLimit: The number of times the router tries to establish the connection before it quits. If this limit is reached, you must restart the connection.
- timeout: Information about the request, which includes the following parameters:

- abandon: An internal timeout in milliseconds for communication between the router and the Application Gateway interface process. If a request exceeds this limit, the router assumes the Application Gateway interface process is off-line.
- late: An internal timeout in milliseconds for communication between the router and the Application Gateway interface process. If a request exceeds this limit, the system software considers the request to be late.
- request: How long in milliseconds the system software waits before timing out a request.

- type: The type of application gateway. The only supported type is CUSTOM_GATEWAY.

Example Get Response

```
<applicationGatewayGlobalSettings>
    <applicationGatewayGlobalSetting>
        <connectionParameters>
            <errorThreshold>10</errorThreshold>
            <heartBeat>
                <interval>15000</interval>
                <limit>10</limit>
                <retry>200</retry>
                <timeout>300</timeout>
            </heartBeat>
            <session>
                <openTimeout>15000</openTimeout>
                <retry>60000</retry>
                <retryLimit>10</retryLimit>
            </session>
            <timeout>
                <abandon>5000</abandon>
                <late>300</late>
                <request>300</request>
            </timeout>
        </connectionParameters>
        <type>CUSTOM_GATEWAY</type>
    </applicationGatewayGlobalSetting>
</applicationGatewayGlobalSettings>
```



CHAPTER 10

Attribute API

Attributes identify a call routing requirement, such as language, location, or agent expertise. You can create two types of attributes: boolean or proficiency. For example, you can create a Boston attribute that specifies that the agent assigned to this attribute must be located in Boston. Then, if a precision queue requires an agent who lives in Boston, then an agent with the attributes Boston = True is a good match. When you create a proficiency attribute, you assign a proficiency level to the agent.

Use the Attribute API to list the attributes currently defined in the database, define new attributes, and view, edit, and delete existing attributes.

URL

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

Operations

- create: Creates an attribute.
- delete: Marks one attribute and associated Agent attribute values for deletion, but does not permanently delete them.
- get: Returns one attribute, using the URL `https://<server>/unifiedconfig/config/attribute/<id>`.
- list: Retrieves a list of attributes.
  - **Query parameters:**
    - selectedAgentCount: Use this query parameter to augment attribute information about multiple agents. The selectedAgentCount parameter shows the number of specified agents associated with this attribute. For example, to find out how many of agents 5000, 5001, 5002, and 5003 in the list have this associated attribute, add `selectedAgentCount=5000,5001,5002,5003`.

Note Using selectedAgentCount automatically sets the summary list query parameter to **true**.

    - Summary list: See list, on page 3.
- update: Updates one attribute.

- **Query Parameters:**
  - removeNonMatchingDepartmentalRefs: Use this query parameter to remove all agent attributes from an attribute when they no longer belong to the department id specified in the query parameter or the global department. For example, to remove all agent attributes that do not belong to department 5000 or the global department, add removeNonMatchingDepartmentalRefs=5000. If this parameter is not specified, the agent attributes must belong to the attribute's department or the global department.

Parameters

- refURL: The refURL of the attribute. See Shared Parameters, on page 8.
- name: The name of the attribute. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- department: A reference to the department (Department API, on page 89), including refURL and name. See References, on page 5.
- dataType: The data type of the attribute. Values are:
  - 3: Boolean.
  - 4: Proficiency.
- defaultValue: Used to specify the default value for the attribute when assigned to an agent, if no explicit value is provided. Values are:
  - Boolean: true\false.
  - Proficiency: 1-10.
- agentAttributes: A collection of agent attribute references for this attribute, including the description, refURL, and read-only parameters agentId, userName, firstName, and lastName. Also includes the attributeValue parameter which indicates the value (true/false or 1-10) of the attribute for this agent. See References, on page 5.
- agentAttributesAdded: A collection of agent attribute references (Attribute API, on page 43) to be added to the attribute, including the agent refURL and the attributeValue of each agent. If the attributeValue is not specified, it is assigned the default value. Agents that already have this attribute are updated with the specified attributeValue. This parameter is update only, and cannot be used in conjunction with the agentAttributes parameter. This parameter can be used with the agentAttributesRemoved parameter. See References, on page 5.
- agentAttributesRemoved: A collection of agent attribute references (Attribute API, on page 43) to be removed from the attribute, including the refURL of each agent. This parameter is update only, and cannot be used in conjunction with the agentAttributes parameter. This parameter can be used with the agentAttributesAdded parameter. See References, on page 5.
- agentCount: Read-only field. Number of agents associated with the attribute.
- selectedAgentCount: Read-only field. Indicates the number of specified agents associated with this attribute. Returned only when using the selectedAgentCount query parameter.

Search and Sort Values

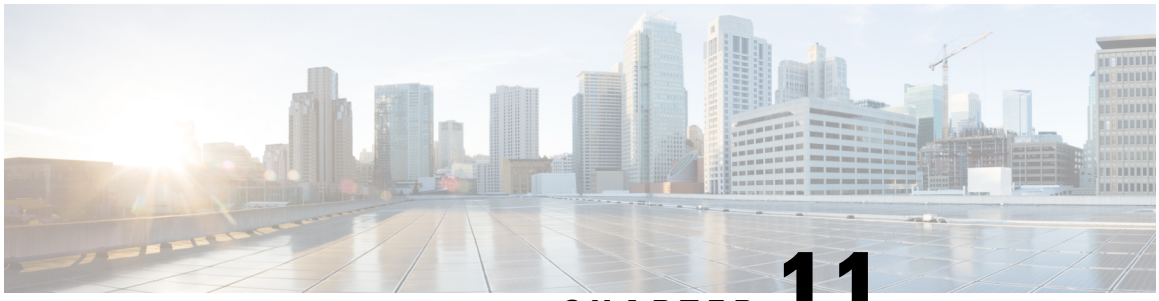
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • dataType • defaultValue • description

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<attribute>

   <refURL>/unifiedconfig/config/attribute/5002</refURL>
         <changeStamp>0</changeStamp>
         <dataType>4</dataType>
         <defaultValue>1</defaultValue>
         <description>test</description>
         <name>AttributeA</name>
</attribute>
```



CHAPTER 11

Bucket Interval API

Configure bucket intervals to report how many calls are handled or abandoned during specific, incremental time slots. Each bucket interval has a maximum of nine configurable time slots, called Upper Bounds. Upper Bounds are ranges measured in seconds to segment and capture call-handling activity. You can run reports that show calls answered and calls abandoned for these intervals.

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.

URL

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

Operations

- create: Creates one bucket interval.
- delete: Deletes one bucket interval from the database.
- get: Returns one bucket interval, using the URL `https://<server>/unifiedconfig/config/bucketinterval/<id>`.
- list: Retrieves a list of bucket intervals.
- update: Updates the name of one bucket interval.

Parameters

- refURL: The refURL of the bucket interval. See Shared Parameters, on page 8.
- name: The name of the bucket interval. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- department: A reference to the department (Department API, on page 89), including the refURL and name. See References, on page 5.
- upperBound1: Required. The first Bucket Interval value, in seconds. Must be greater than 0. This parameter cannot be updated.

- upperBound2 to upperBound 9: Optional. The next Bucket Interval values, in seconds. Each must be greater than the previous upperBound field or be left blank (if blank, all remaining upperBound fields must also be blank). These parameters cannot be updated.

Search and Sort Values

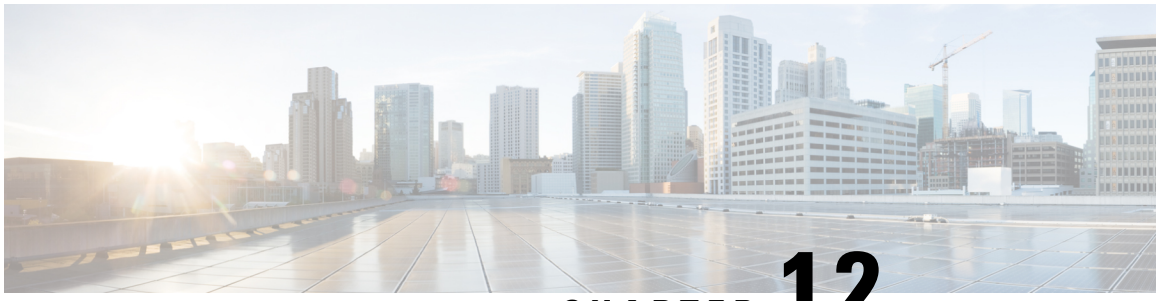
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name	• name (default) • upperBound 1-9

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<bucketInterval>
    <refURL>/unified/config/bucketInterval/(id)</refURL>
    <name>test</name>
    <department>
        <name>sales</name>
        <refURL>/unifiedconfig/config/department/5003</refURL>
    </department>
    <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>
```



CHAPTER 12

Bulk Job API

Bulk jobs are a fast and efficient way to enter data at initial setup and to incorporate large-scale changes, such as changing agent skill groups between shifts and hiring multiple new agents.

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.

URL

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

Operations

- create: Creates one bulk job.
- delete: Permanently deletes one bulk job.
- get: Returns one bulk job using the URL `https://<server>/unifiedconfig/config/bulkjob/<id>`.
- get (template): Returns a sample CSV template for the given bulk job type. The response contains the CSV template as a file attachment. The form of the request is `https://<server>/unifiedconfig/config/bulkjob/templates/<jobType>`, where <jobType> can be one of the following:
  - dialednumber
  - agent
  - calltype
  - skillgroup
  - inventory (for lab mode inventory)
  - CVP file transfer
  - ssomigration

    This template returns records for agent and supervisor accounts that are not enabled for single sign-on. You can use the template to migrate these user accounts to SSO.

    Non-SSO users can be filtered by peripheralId, agentTeamId, or both.

To filter by peripheralId, use the following URL:
`https://<server>/unifiedconfig/config/bulkjob/templates/ssomigration?q=peripheralId:<peripheralId>`.

To filter by agentTeamId, use the following URL:
`https://<server>/unifiedconfig/config/bulkjob/templates/ssomigration?q=agentTeamId:<agentTeamId>`.

To filter by both peripheralId and agentTeamId, use the following URL:
`https://<server>/unifiedconfig/config/bulkjob/templates/ssomigration?q=peripheralId:<peripheralId> agentTeamId:<agentTeamId>`.

- list: Retrieves a list of bulk jobs.

Parameters

- refURL: The refURL of the bulk job. See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- fileContent: The content of the bulk CSV file. The size of the file must not exceed 3MB. For information about the CSV file data, see the "Manage Bulk Jobs" section of the *Cisco Packaged Contact Center Enterprise Administration and Configuration Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html.
- createDateTime: The time the bulk job was submitted. It indicates the time in milliseconds elapsed from the zero epoch value of January 1, 1970, 00:00:00 GMT. Read-only.
- jobHostName: The Windows computer name of the AW that initiated the bulk job. Read-only.
- startDateTime: The time the bulk job began executing. Read-only.
- endDateTime: The time the bulk job completed or failed. Read-only.
- jobState: The current state of the job. Read-only.
  - 1: Queued
  - 2: Processing
  - 3: Succeeded
  - 4: Failed
  - 5: Cancelled
  - 6: Partially succeeded
- jobType: The job type. Optional. If this parameter is not provided, the value is determined automatically based on the header in the bulk job file. If the parameter is provided, the header must match the job type specified.
  - 1: Dialed Number
  - 2: Agent
  - 3: Call Type

- 4: Skill Group
- 5: SSO Migration
- 5: SSO Migration
- 6: IVR Applications
- 102: Inventory

- logFile: A URL to download the log file for the bulk job. Read-only.
- csvFile: A URL to download the CSV file that was originally uploaded in the fileContent parameter. Read-only.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

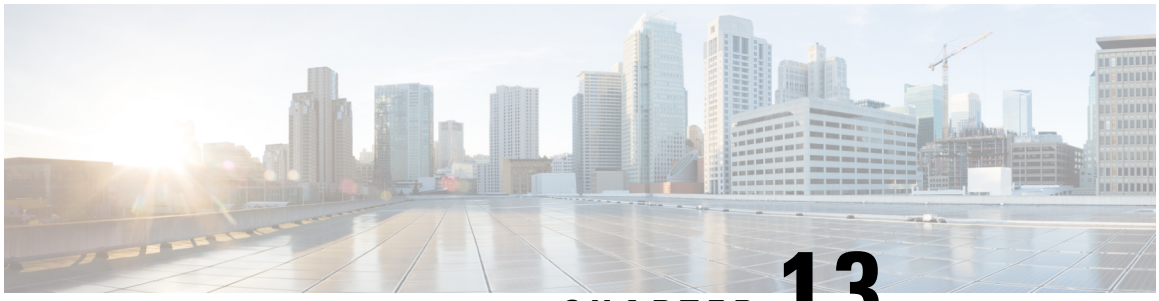
Search parameters	Sort parameters
• description	• description • jobType • jobState • jobHostName • createDateTime • startDateTime • endDateTime

See Search, on page 10 and Sort, on page 11.

Example Get Response

Example response for a dialed number create job that completed successfully:

```
<bulkJob>
  <changeStamp>0</changeStamp>
  <refURL>/unifiedconfig/config/bulkjob/(id)</refURL>
  <jobHostName>CCE-AW-1</jobHostName>
  <createDateTime>1330441858360</createDateTime>
  <startDateTime>1330441858361</startDateTime>
  <endDateTime>1330441858368</endDateTime>
  <jobState>3</jobState>
  <jobType>1</jobType>
  <description>dn create bulk job</description>
  <logFile><refURL>/unifiedconfig/config/bulkjob/(id)/log</refURL></logFile>
  <csvFile><refURL>/unifiedconfig/config/bulkjob/(id)/csv</refURL></csvFile>
</bulkJob>
```



CHAPTER 13

Business Hours API

- Business Hours API, on page 53
- Business Hour Status Reason API, on page 57
- Time Zone API, on page 58

Business Hours API

Use this API to list the currently defined business hours, define new business hours, and view, edit, and delete the existing business hours.

URL

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

Operations

- create: Creates one business hour.
- get: Returns one business hour using the URL `https://<server>/unifiedconfig/config/businesshour/<id>`
- get(template): Returns a sample CSV template for Special Hours and Holidays, which is provided by the API, using the URL `https://<server>/unifiedconfig/config/businesshour/specialdayschedule/template`. The response contains the CSV template as a file attachment.
- list: Retrieves a list of business hours.
- update: Updates one business hour.
- delete: Permanently deletes one business hour.

Parameters

- changeStamp: See Shared Parameters, on page 8.
- refURL: The refURL of the business hour. See Shared Parameters, on page 8.
- name: The name of the business hour. See Shared Parameters, on page 8.

- description: See Shared Parameters, on page 8.
- type: Required. The type of the business hour.
  - 0: 24x7
  - 1: Custom
- department: Optional. A reference to the department associated with the business hour, including the refURL. See References, on page 5.
- configuredStatus: The configured status of the business hour.
  - status: Required.
    - 0: Calendar Schedule
    - 1: Force Close
    - 2: Force Open
  - statusReason: Required if the status is Force Open or Force Close.
- runTimeStatus: The run time status of the business hour. Available only in Get. It cannot be set or updated.
- timezone: Required. The time zone of the business hour.
- weekDaySchedules: The list of schedules on weekdays of business. Required only when business hour type is Custom.
  - dayOfweek: Required.
    - 0: Sunday
    - 1: Monday
    - 2: Tuesday
    - 3: Wednesday
    - 4: Thursday
    - 5: Friday
    - 6: Saturday
  - startTime: Required. Format HH:MM
  - endTime: Required. Format HH:MM
- specialDaySchedules: The list of schedules on special days of business.
  - date: Required. Format: DD-MM-YYYY
  - startTime: Required, if the status is Open. Format: HH:MM
  - endTime: Required, if the status is Open. Format: HH:MM
  - description: Optional. Maximum of 255 characters.

- status: Required.
  - 0: Closed
  - 1: Open
- statusReason: Required. The refURL to existing status reason.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name • runTimeStatus • runTimeStatusReason • type • department.name • timezone.dispName

See Search, on page 10 and Sort, on page 11.

Advanced Search Parameter

- **departments: (dept1|dept2|dept3...)** returns all the business hours which belong to any of the specified departments. You can specify up to 10 departments. The department name is fully matched (case-insensitive, no partial matches).

Example Get Response

```
<businessHour>
    <refURL>/unifiedconfig/config/businesshour/1000</refURL>
    <changeStamp>0</changeStamp>
    <runTimeStatus>2</runTimeStatus>
    <runTimeStatusReason>closed</runTimeStatusReason>
    <name>credit_card</name>
    <type>1</type>
    <description>Business Line of Credit</description>
    <timezone>
    <refURL>/unifiedconfig/config/timezone/3001</refURL>
    <displayName>(UTC-05:00) Eastern Time (US & Canada)</displayName>
    </timezone>
    <configuredStatus>
        <status>1</status>
        <statusReason>
            <refURL>/unifiedconfig/config/businesshourstatusreason/5000</refURL>
            <reasonText>Emergency</reasonText>
        </statusReason>
    </configuredStatus>
    <specialDaySchedules>
        <specialDaySchedule>
          <refURL>/unifiedconfig/config/businesshour/1000/specialdayschedule/2001</refURL>
```

Business Hours API

```
            <changeStamp>0</changeStamp>
            <date>04-02-2019</date>
            <startTime>03:26</startTime>
            <endTime>04:16</endTime>
            <description>Thanksgiving Day</description>
            <status>1</status>
            <statusReason>
                <refURL>/unifiedconfig/config/businesshourstatusreason/5000</refURL>
                <reasonText>Emergency</reasonText>
            </statusReason>
        </specialDaySchedule>
    </specialDaySchedules>
    <weekDaySchedules>
        <weekDaySchedule>
            <refURL>/unifiedconfig/config/businesshour/1000/weekdayschedule/1001</refURL>
            <changeStamp>0</changeStamp>
            <dayOfWeek>1</dayOfWeek>
            <startTime>06:20</startTime>
            <endTime>07:22</endTime>
        </weekDaySchedule>
    </weekDaySchedules>
</businessHour>

<businessHour>
    <refURL>/unifiedconfig/config/businesshour/1000</refURL>
    <changeStamp>0</changeStamp>
    <runTimeStatus>2</runTimeStatus>
    <runTimeStatusReason>Week Day open reason</runTimeStatusReason>
    <name>Businesshour</name>
    <type>1</type>
    <description>additional requirement</description>
    <department>
        <refURL>/unifiedconfig/config/department/5022</refURL>
        <name>debit_card</name>
    </department>
    <timezone>
        <refURL>/unifiedconfig/config/timezone/3001</refURL>
        <displayName>(UTC-08:00) Pacific Time (US & Canada)</displayName>
    </timezone>
    <configuredStatus>
        <status>1</status>
        <statusReason>
            <refURL>/unifiedconfig/config/businesshourstatusreason/5000</refURL>
            <reasonText>Emergency</reasonText>
        </statusReason>
    </configuredStatus>
    <specialDaySchedules>
        <specialDaySchedule>
           <refURL>/unifiedconfig/config/businesshour/1000/specialdayschedule/2001</refURL>

            <changeStamp>0</changeStamp>
            <date>04-02-2019</date>
            <startTime>03:26</startTime>
            <endTime>04:16</endTime>
            <description>Thanksgiving Day</description>
            <status>1</status>
            <statusReason>
                <refURL>/unifiedconfig/config/businesshourstatusreason/5000</refURL>
                <reasonText>Week Day open reason</reasonText>
            </statusReason>
        </specialDaySchedule>
    </specialDaySchedules>
    <weekDaySchedules>
        <weekDaySchedule>
            <refURL>/unifiedconfig/config/businesshour/1000/weekdayschedule/1001</refURL>
```

```
            <changeStamp>0</changeStamp>
            <dayOfWeek>1</dayOfWeek>
            <startTime>06:20</startTime>
            <endTime>07:22</endTime>
        </weekDaySchedule>
    </weekDaySchedules>
</businessHour>
```

Business Hour Status Reason API

Use this API to list the currently defined business hour status reasons, define new status reasons, and view, edit, and delete the existing status reasons.

URL

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

Operations

- create: Creates one business hour status reason.
- get: Returns one business hour status reason using the URL `https://<server>/unifiedconfig/config/businesshourstatusreason/<id>`
- list: Retrieves a list of business hour status reasons.
- update: Updates one business hour status reason.
- delete: Permanently deletes one business hour status reason.

Parameters

- category: Optional. The category of the reason codes: User-defined or System-defined.
- reasonCode: Required. The unique status reason code for business hour. Range is 1001 to 65535.

Note Codes 1 to 1000 are reserved as system defined reason codes. The System-defined reason codes cannot be deleted.

- reasonText: Optional. The reason for the business schedule. Maximum of 255 characters.

Example Create Request

```
<businessHourStatusReason>
 <category>User-defined</category>
<reasonCode>1234</reasonCode>
<reasonText>Emergency</reasonText>
</businessHourStatusReason>
```

Example Get Response

```
<businessHourStatusReason>
 <refURL>/unifiedconfig/config/businesshourstatusreason/5000</refURL>
 <changeStamp>0</changeStamp>
 <category>User-defined</category>
 <reasonCode>2550</reasonCode>
 <reasonText>Open</reasonText>
 </businessHourStatusReason>
```

Time Zone API

Use the Time Zone API to list all available time zones and to get time zone information for a specified zone. Time zone information is stored in the registry of the Windows operating system.

Important Microsoft periodically releases cumulative time zone updates. These updates include worldwide changes to time zone names, bias (the amount of time in minutes that a time zone is offset from Coordinated Universal Time (UTC)), and observance of daylight saving time. These patches update the information in the Windows registry. When these updates are available, apply them to all virtual machines in the deployment that are running a Microsoft Windows operating system.

Use this API with the Business Hours API to set the default time zone for a business hour.

This API is read-only.

URL

`https://<server>/unifiedconfig/config/timezone/v2`

Operations

- list: Retrieves a list of available time zones. The list is sorted by UTC offset from the International Date Line from west to east.
- get: Returns information for a specific time zone using the URL `https://<server>/unifiedconfig/config/timezone/v2/<id>`.

Response Parameters

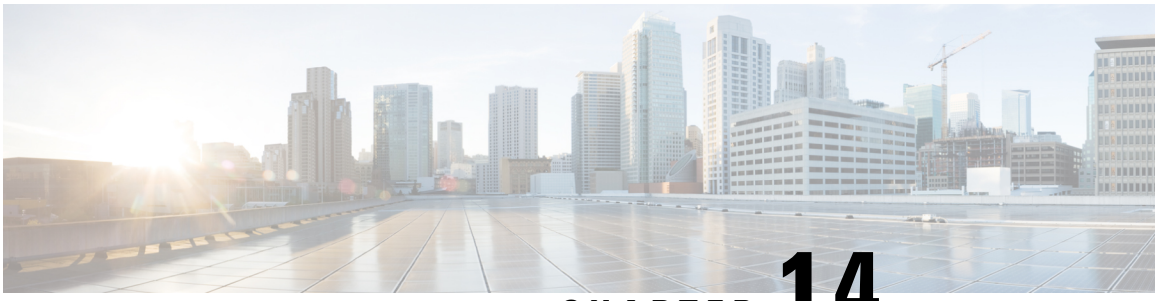
- name: The name of the time zone.
- displayName: Specific bias and location information about the time zone, such as the offset from UTC and one or more places located within the time zone.

  Example: “(UTC-05:00) Eastern Time (US & Canada)”
- changeStamp: See Shared Parameters, on page 8.

Example Get Response

```
<timezone>
    <refURL>/unifiedconfig/config/timezone/v2/5000</refURL>
    <changeStamp>0</changeStamp>
```

```
    <displayName>(UTC-05:00) Eastern Time (US & Canada)</displayName>
    <name>UTC</name>
</timezone>
```



CHAPTER 14

Call Type API

Call types categorize calls. Based on call type, the system maps a dialed number (see Dialed Number API, on page 95) to a routing script that ultimately sends the call to the appropriate destination.

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.

URL

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

Operations

- create: Creates one call type.
- delete: Marks one call type for deletion, but does not permanently delete it.
- get: Returns one call type, using the URL `https://<server>/unifiedconfig/config/calltype/<id>`.
- list: Retrieves a list of call types.
- update: Updates one call type.

Parameters

- refURL: The refURL of the call type. See Shared Parameters, on page 8.
- name: The name of the call type. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- department: A reference to the department (Department API, on page 89), including the refURL and name. See References, on page 5.
- id: The database id of the call type. Read-only field. Used in scripting.
- serviceLevelThreshold: Maximum time in seconds that a caller should wait before being connected with an agent. Leave blank to use the system default (set in the Global API, on page 111).
- serviceLevelType: This value indicates how the system calculates the service level.

- blank: Use the system default.
- 1: Ignore Abandoned Calls.
- 2: Abandoned Calls have Negative Impact.
- 3: Abandoned Calls have Positive Impact.

- bucketInterval: A reference to the bucket interval (Bucket Interval API, on page 47), including the refURL and name.
- survey: A reference to the Survey Survey API including the refURL and name.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description • id	• name (default) • description • id • serviceLevelThreshold • serviceLevelType

See Search, on page 10 and Sort, on page 11.

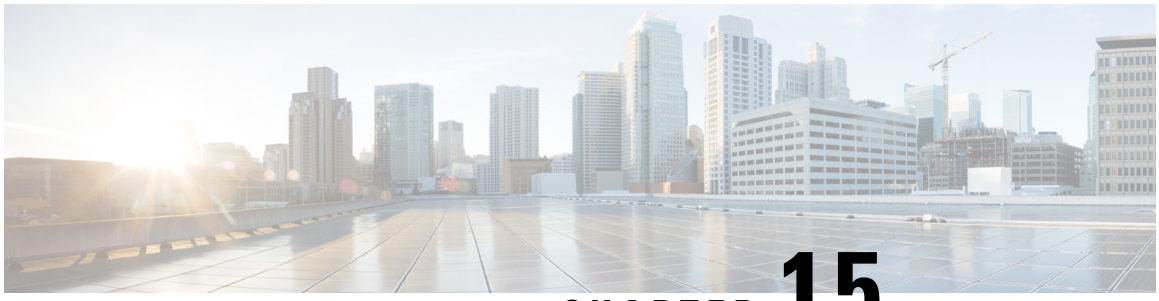
Example Get Response

```
 <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>
```

Example Get Response

```
 <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>
    <name>BuiltIn</name>
    <survey>
       <refURL>/unifiedconfig/config/survey/2</refURL>
```

```
    </survey>
</callType>
```



CHAPTER 15

Congestion Control API

Congestion control parameters determine how calls are treated by the system when too many calls are received at one time. Use the Congestion Control API to list the current congestion control parameters in the database.

URL

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

Operations

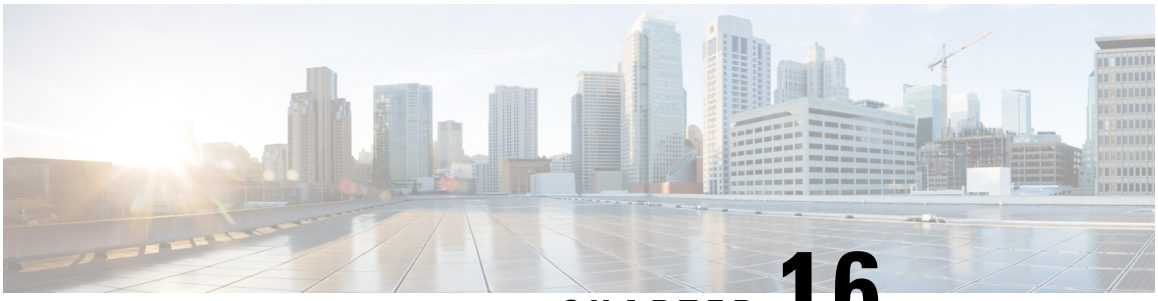
- get: Returns the congestion control parameters, using the URL `https://<server>/unifiedconfig/config/congestioncontrol`.

Parameters

- deploymentType: The type of deployment. See Deployment Type Info API, on page 93.
- congestionEnabled: Indicates if congestion control is enabled. Value is true/false.
- congestionTreatmentMode: Mode to handle congestion. Values are:
  - 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. Only used if congestionTreatmentMode is set to 3 (Treat call with System default label).
- cpsCapacity: The maximum number of calls per second allowed.
- cpsCapacityDefault: The default value for the cpsCapacity parameter for the current deployment type. Read-only.

Example Get Response

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



CHAPTER 16

Context Service Configuration API

Use the Context Service Configuration API to configure Context Service settings for CVP, Finesse, and SocialMiner.

URL

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

Operations

- get: Gets the Context Service configuration data, using the URL `https://<server>/unifiedconfig/config/contextservice`.
- update: Updates the Context Service configuration data.

Parameters

- proxyUrl: URL of the proxy server. The default is an empty string. Optional.
- maxRetries: The maximum number of times that the system attempts the operation. This parameter must be set to 1.
- timeout: The amount of time, in milliseconds, that the system waits before abandoning the attempt to perform the operation. The default is 1200 ms. The values are 200 - 15000 ms. Optional.
- labMode: Whether the Context Service is in lab mode. The default is false. Optional.

Example Get Response

```
 <contextServiceConfig>
   <proxyUrl>http://example.com:80</proxyUrl>
   <maxRetries>1</maxRetries>
   <timeout>10000</timeout>
   <labMode>false</labMode>
</contextServiceConfig>
```

Following are the REST responses received during execution of REST API to configure the Context Service for CVP or Finesse:

- **Success** - Configuration changes persist in AW DB and synchronized with respective devices.

  Code: 200

Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with one or more devices of CVP and/or Finesse.

  Code: 201

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with CVP or Finesse failed.)

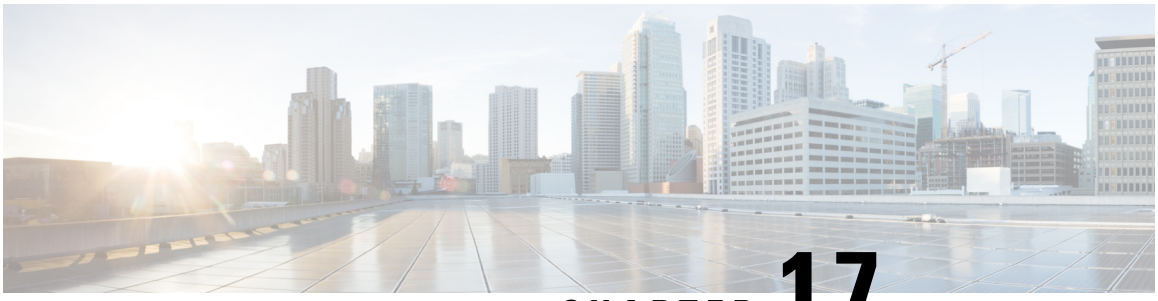
  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

```
<apiErrors>
<apiError>
<errorMessage>Configuration update failed for one or more devices.</errorMessage>
<errorType>PARTIAL_SUCCESS</errorType>
</apiError>
</apiErrors>
```

- **Failure**- The configuration updates to AW DB is failed.



CHAPTER 17

Context Service Registration API

Cisco Context Service is a cloud-based omnichannel solution for Cisco Contact Center Enterprise Solutions. It enables you to capture your customer's interaction history by providing flexible storage of customer-interaction data across any channel.

Various components in the CCE Solution provide out of the box integration with Context Service. Context Service also provides an API for integration with your own applications or third-party applications to capture end-to-end customer-interaction data.

For more information about Context Service, see *Cisco Packaged Contact Center Enterprise Features Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html

Use the Context Service Registration API to register CVP, Finesse, and SocialMiner with Context Service and get the current registration status.

URL

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

Operations

- get: Returns the current Context Service registration state using the URL `https://<server>/unifiedconfig/config/contextserviceregistration?callbackUrl=(url)`.
- update: Updates Context Service connection data in CVP, Finesse, or SocialMiner and the CCE database.

Parameters

- callbackUrl: This parameter is optional in the Get URL. The browser is redirected to this URL to complete the registration or deregistration process.
- state: The Context Service registration state. Values are as follows:
  - notRegistered
  - registered
  - invalid: The connection data used to register with Context Service is no longer valid. Reregister to recover from this state.

- managementUrl: Navigate to this URL to manage the Context Service. This parameter is returned if the state is registered.
- registrationUrl: Navigate to this URL to begin Context Service registration. This parameter is returned if you include the callbackUrl in the Get URL and the state is either notRegistered or invalid.
- deregistrationUrl: Navigate to this URL to begin Context Service deregistration. This parameter is returned if you include the callbackUrl in the Get URL and the state is registered.
- refURL: The refURL. See Shared Parameters, on page 8.
- connectionData: This parameter is required and the maximum size is 4 KB. connectionData is an opaque string used to connect to the Context Service. Use an empty string to clear the connectionData when deregistering.

Note You must deregister before reregistering to Context Service. If the current connectionData is not empty, you can set the value either to an empty string or to the current value.

Sample API Workflow

1. The user navigates to a custom Context Service web page.
2. The page calls the Get API to get the current registration status from Context Service. The page displays either a Register or Deregister button.
3. The user clicks Register or Deregister.
4. The page calls the Get API with a callback URL for a custom web page. This page can be the custom Context Service web page from step 1. The API returns either the Context Service registration or deregistration URL.
5. The browser redirects to the Context Service registration or deregistration URL.
6. The user completes the registration or deregistration wizard.
7. The browser redirects back to the callback URL with a connectionData query parameter appended to it.
8. The callback page calls the Update API with the connection data from step 7 to send the data to CVP, Finesse, or SocialMiner and the CCE database.
9. Go to step 2.

Example Get Response

```
 <contextServiceRegistration>
   <state>registered</state>
   <managementUrl>(url)</managementUrl> <!-- Included if state is registered -->
   <registrationUrl>(url)</registrationUrl> <!-- Included if state is notRegistered
    and callbackUrl is provided -->
   <deregistrationUrl>(url)</deregistrationUrl> <!-- Included if state is registered
    and callbackUrl is provided -->
   <refURL>/unifiedconfig/config/contextservice</refURL>
</contextServiceRegistration>
```

Example Update Request

```
 <contextServiceRegistration>
   <connectionData>(data)</connectionData>
</contextServiceRegistration>
```

Following are the REST responses received during execution of REST API to register the Context Service with CVP or Finesse:

- **Success** - Configuration changes persist in AW DB and synchronized with respective devices.

  Code: 200

  Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with one or more devices of Finesse and/or CVP.

  Code: 201

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with CVP or Finesse failed.)

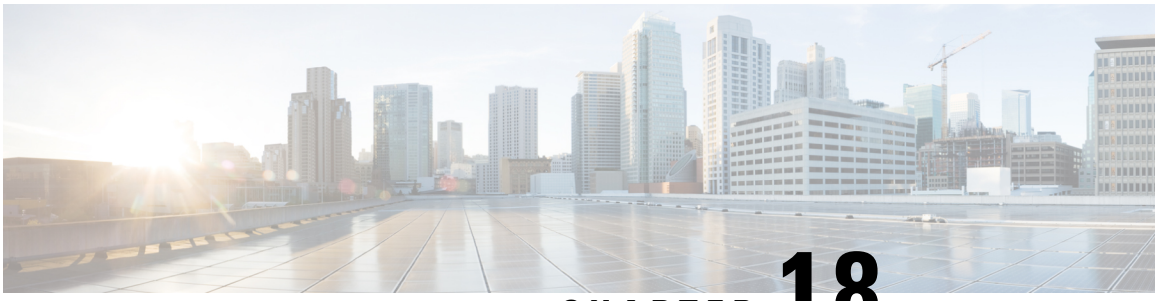
  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

  ```
  <apiErrors>
  <apiError>
  <errorMessage>Configuration update failed for one or more devices.</errorMessage>
  <errorType>PARTIAL_SUCCESS</errorType>
  </apiError>
  </apiErrors>
  ```

- **Failure**- The configuration updates to AW DB is failed.



CHAPTER 18

Courtesy CallBack API

The Courtesy Callback (CCB) feature is available in Unified CVP. Courtesy Callback reduces the time callers have to wait on hold/in queue. The feature allows the system to offer callers who meet certain criteria, for example, callers with the possibility of being in queue for more than X minutes, the option to be called back by the system when the wait time would be considerably shorter.

Courtesy CallBack feature of CVP can be configured through PCCE UI. The configured data persist in AW DB and then synchronized to the CVP Call Servers of the site through the CCB API.

URL

To associate the Unified CVP Reporting Server. The Unified CVP Reporting Server IP address is saved in AW DB, and synchronized to realtimedb.properties of all the CVP Call Servers present in a site.
`https://<server>/unifiedconfig/config/ccb/{datacenter}`

Operations

- get: Returns the IP address or hostname of the CVP Reporting Server, using the URL: `https://<server>/unifiedconfig/config/ccb/{datacenter}`.
- update: Updates the IP address or hostname of the CVP Reporting Server.

Parameters

- reportingAddress: Required. IP address or hostname of the Cisco Unified CVP Reporting Server in PCCE Inventory. The value can be null or empty. If the value is not empty, you must provide a valid IPv4 or IPv6 address.

Example Get Response - CVP Call Server

```
<?xml version="1.0" encoding="UTF-8"?>
<CCB>
<reportingAddress>10.10.10.10</reportingAddress>
</CCB>
```

Following are the REST responses received during the execution of REST API to configure the Courtesy CallBack:

- **Success** - Configuration changes persist in AW DB and synchronized with CVP Call Server.

  Code: 200

Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with CVP Call Server.

  Code: 201

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with CVP failed.)

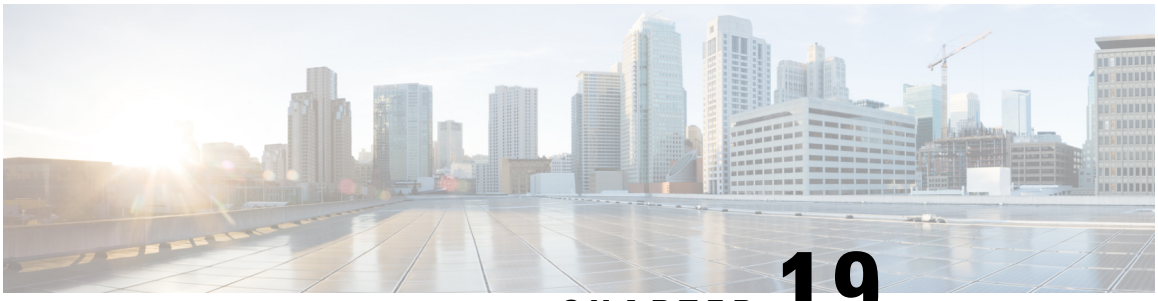
  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

```
<apiErrors>
<apiError>
<errorMessage>Configuration update failed for one or more devices.</errorMessage>
<errorType>PARTIAL_SUCCESS</errorType>
</apiError>
</apiErrors>
```

- **Failure**- The configuration updates to AW DB is failed.



CHAPTER 19

CVP Device Configuration API

Use the GET API to get the device configuration information of the CVP Call Server for the specified remote site. If no remote site is specified, it will retrieve the information for the main site.

URL

`https://<server>:<serverport>/unifiedconfig/config/deviceconfig/<site_name>/cvpserver`

Operations

- get: Gets the device configuration information of the CVP Call Server for the specified remote site using the URL: `https://<server>:<serverport>/unifiedconfig/config/deviceconfig/<site_name>/cvpserver`.
- update: Updates the configuration details for CVP devices.

  If the configuration types (ICM, SIP, IVR, VXML, Infra, Media Server) are set as empty, the configuration related to the respective configuration type gets deleted.

Parameters

- **ICM**
  - vruConnectionPort: Required. The port number on which the ICM Service listens for a TCP connection from the ICM PIM. It can be any valid TCP/IP connection port.
  - : Required. The maximum length of an incoming Dialed Number Identification Service (DNIS). The range is 1 - 99999.
  - enableTrunkUtilization: Default value is false.
  - maxGatewayPorts: Maximum number of ports that a gateway supports. This is available only if enableTrunkUtilization is true. The range is 1 - 1500.
  - gateways: The list of gateways available for trunk reporting. This is available only if enableTrunkUtilization is true.
    - gateway: The gateway available for trunk reporting.
      - refURL: The MachineInventory reference URL of the gateway machine.
      - trunkGroupId: The Trunk Group ID of the gateway.

- enableSecureVRU: To enable secure communication between ICM and VRU PIM. Default value is false.

- **SIP**
  - enableOutboundProxy: Use a Cisco Unified SIP Server Group.
  - proxyHost: The SIP Server Group name of type External. This is available only if useOutboundProxy is true.
  - outboundProxyPort: Port on which the SIP service sends requests to the outbound proxy server. This is available only if useOutboundProxy is true. The range is 1-65535.
  - useSrv: Use DNS SRV for outbound proxy lookup.
  - useLocalSrv: Select to resolve the SRV domain name with a local configuration file instead of a DNS Server. Values can be enabled/disabled.
  - outgoingTransport: Transport type for outgoing SIP requests. It can be TCP/UDP.
  - incomingPort: Required. Port on which the SIP Service listens for incoming SIP requests. The range is 1-65535.
  - sigDigits: The range is 1-20.
  - useErrorRefer: Default is false.
  - toneDuration: Required. Default is 100.
  - commaDuration: Required. Default is 100.
  - sipHeaders: It can have a maximum length of 255 characters.
    - sipHeader
      - name
      - value
  - incomingSecurePort: Required. Default is 5061.
  - incomingSecureTransport: Default is TLS.
  - outgoingSecureTransport: Default is TLS.
  - supportedTLSVersions: Default is TLSv1.2. Values are TLS1.0, TLSv1.1, and TLSv1.2.
  - secureCiphers: Default is TLS_RSA_WITH_AES_128_CBC_SHA.

- **IVR**
  - generateHttpsURL: Default is yes. Values can be true/false or yes/no.
  - useBackupMediaServers, useBackupASRTTSServers: If you select Yes (default) and a media server is unavailable, the gateway attempts to connect to the backup Media Server. Values can be true/false or yes/no.

- useMediaServerHostNames: If you select No (default), the IP address is used for the XML Server and Media Server. If you select Yes, the hostnames are used rather than IP addresses. Values can be true/false or yes/no.
- CallTimeout: Required. The number of seconds the IVR Service waits for a response from the SIP Service before timing out. This setting should be longer than the longest prompt, transfer or digit collection at a Voice Browser. If the timeout is reached, the call is canceled but no other calls are affected. The only downside to making the number arbitrarily large is that if calls are being stranded, they will not be removed from the IVR Service until this timeout is reached. Default is 7200. It can be 6 seconds or higher.

- **VXML**
  - enableReporting: Indicates whether or not the Unified CVP VXML Server sends data to the Reporting Server. If disabled, no data is sent to the Reporting Server, and reports do not contain any VXML application data. Value can be enabled/disabled.
  - reportingDetail: Indicates whether VXML application details are reported. Value can be enabled/disabled.
  - inclusiveFilters: List of applications, element types, element names, and element fields, and ECC variables to include in reporting data. Range is a semicolon-separated list of text strings. A wildcard character (*) is allowed within each element in the list.
  - exclusiveFilters: List of applications, element types, element names, and element fields, and ECC variables to exclude from reporting data. Range is a semicolon-separated list of text strings. A wildcard character (*) is allowed within each element in the list.

- **Infra**
  - maxLogFileSize: Required. Maximum number of threads allocated in the thread pool, which can be shared by all services running as part of a CVP Web Application. Range can be 100-1000.
  - maxLogDirectorySize: Required. Maximum number of megabytes to allocate for disk storage for log files. Note Modifying the value to a setting that is below the default value might cause logs to be rolled over quickly. Consequently, log entries might be lost, which can affect troubleshooting. Range is 500 - 500000. The log folder size divided by the log file size must be less than 5000.
  - syslogServer: Hostname or IP address of Primary Syslog Server to send syslog events from a CVP Application. It can be a valid IP address or hostname.
  - syslogServerPort: Port number of Primary Syslog Server. The range is 1-65535.
  - syslogBackupServer: Hostname or IP address of the Primary Backup Syslog Server to send syslog events from a CVP Application when the Syslog Server cannot be reached. It can be a valid IP address or hostname.
  - syslogBackupServerPort: Port number of Primary Backup Syslog Server. The range is 1-65535.
  - syslogSecondaryServer: Hostname or IP address of the Secondary Syslog Server to send syslog events from a CVP Application when the Syslog Server cannot be reached. It can be a valid IP address or hostname.
  - syslogSecondaryServerPort: Port number of Secondary Syslog Server. The range is 1-65535.

- syslogSecondaryBackupServer: Hostname or IP address of the Secondary Backup Syslog Server to send syslog events from a CVP Application when the Syslog Server cannot be reached. It can be a valid IP address or hostname.
- syslogSecondaryBackupServerPort: Port number of Secondary Backup Syslog Server. The range is 1-65535.

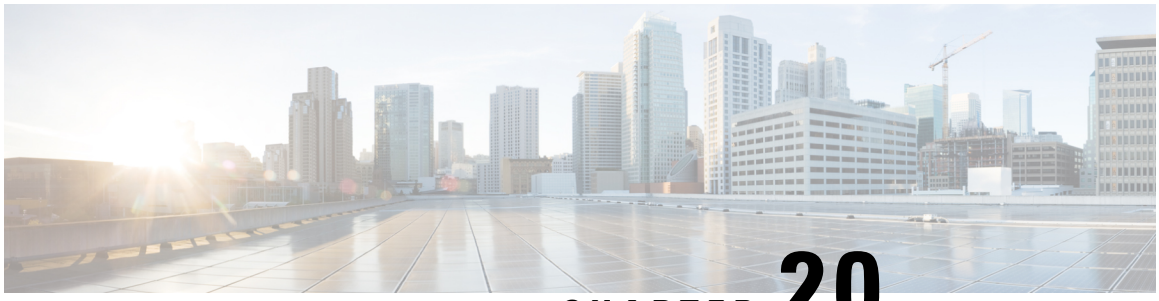
- **Media Server Properties**
  - defaultMediaServer: To set the default Media Server for all the available CVP Servers in the specified site. Provide IP address of the Media Server.

Note
- You must configure atleast one Media Server.
- Default Media Server must be one of the configured Media Servers.

Example Get Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CVP>
 <icm>
   <enableSecureVRU>false</enableSecureVRU>
   <enableTrunkUtilization>false</enableTrunkUtilization>
   <gateways/>
   <maxDNISLength>10</maxDNISLength>
   <maxGatewayPorts>700</maxGatewayPorts>
   <vruConnectionPort>5000</vruConnectionPort>
 </icm>
 <infrastructure>
   <maxLogDirectorySize>20000</maxLogDirectorySize>
   <maxLogFileSize>100</maxLogFileSize>
   <syslogBackupServer/>
   <syslogBackupServerPort>514</syslogBackupServerPort>
   <syslogSecondaryBackupServer/>
   <syslogSecondaryBackupServerPort>514</syslogSecondaryBackupServerPort>
   <syslogSecondaryServer/>
   <syslogSecondaryServerPort>514</syslogSecondaryServerPort>
   <syslogServer/>
   <syslogServerPort>514</syslogServerPort>
 </infrastructure>
 <ivr>
   <callTimeout>7200</callTimeout>
   <useBackupMediaServers>true</useBackupMediaServers>
   <useMediaServerHostNames>false</useMediaServerHostNames>
   <useSecurityForMediaFetches>false</useSecurityForMediaFetches>
 </ivr>
<mediaServer>
   <defaultMediaServer>10.10.10.100</defaultMediaServer>

</mediaServer>
<sip>
  <commaDuration>100</commaDuration>
  <enableOutboundProxy>false</enableOutboundProxy>
  <incomingPort>5060</incomingPort>
  <incomingSecurePort>5061</incomingSecurePort>
  <outboundProxyPort>5060</outboundProxyPort>
  <outgoingTransport>TCP</outgoingTransport>
```

```
  <secureCiphers>TLS_RSA_WITH_AES_128_CBC_SHA</secureCiphers>
  <sigDigits>0</sigDigits>
  <sipHeaders>
   <sipHeader>
     <name>n1</name>
     <value>v1</value>
   </sipHeader>
  <sipHeader>
     <name>n1</name>
     <value>v1</value>
  </sipHeader>
 </sipHeaders>
 <supportedTLSVersion>TLSv1.2</supportedTLSVersion>
 <toneDuration>100</toneDuration>
 <useErrorRefer>true</useErrorRefer>
 <useLocalSrv>true</useLocalSrv>
 <useSrv>true</useSrv>
</sip>
<vxml>
  <enableAppDetailsReporting>false</enableAppDetailsReporting>
  <enableReporting>true</enableReporting>
</vxml>
</CVP>
```



CHAPTER 20

CVP DNIS API

- CVP Dialed Number Identification Service API , on page 81

CVP Dialed Number Identification Service API

Use the CVP DNIS API to retrieve and update the DNIS number in the CVP call server.

URL

```
https://<server>:<serverport>/unifiedconfig/config/deviceconfig/cvpdnis
```

Operations

- **Get**:

  Retrieves the CVP DNIS list of a single CVP call server. To retrieve the DNIS list, enter the hostname of the CVP call server in the URL
  `https://<server>:<serverport>/unifiedconfig/config/deviceconfig/cvpdnis?hostname=<hostname>`

  Note Hostname is a mandatory parameter in the Get URL.

- **Put**:

  Updates the CVP DNIS list of a single CVP call server. To update the DNIS list, enter the hostname of the CVP call server in the payload.
  `https://<server>:<serverport>/unifiedconfig/config/deviceconfig/cvpdnis`

Parameters

- hostname: Indicates the CVP hostname as defined in the PCCE inventory
- dnisList: Indicates the semi-colon separated list of DNIS numbers. Enter a DNIS number or DNIS range.

  The length of each DNIS in this list can be up to 32 characters. DNIS must be a positive integer; DNIS can begin with a zero (0).

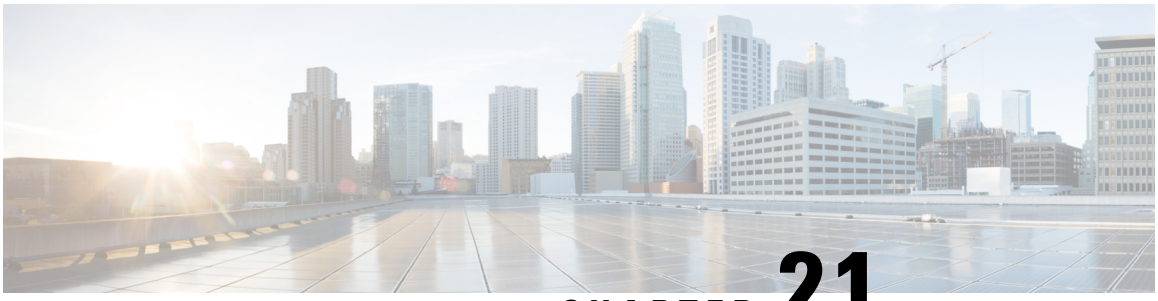
  DNIS range is 1- 32 characters. The upper and lower limit of the DNIS range must be of the same length. For example, a range from 100 - 900 is valid because each number is three characters in length.

Example Get Response

```
<cvpDnis>
    <hostname>cvp1a.berlin.icm</hostname>
    <dnisList>12312355;12412455</dnisList>
</cvpDnis>
```

Responses

- **Success**: 200 OK. The DNIS number was successfully updated in the CVP call server.
- **Errors**: Errors indicate that the validation has failed. Check the error messages to enter the correct hostname or dnisList.



CHAPTER 21

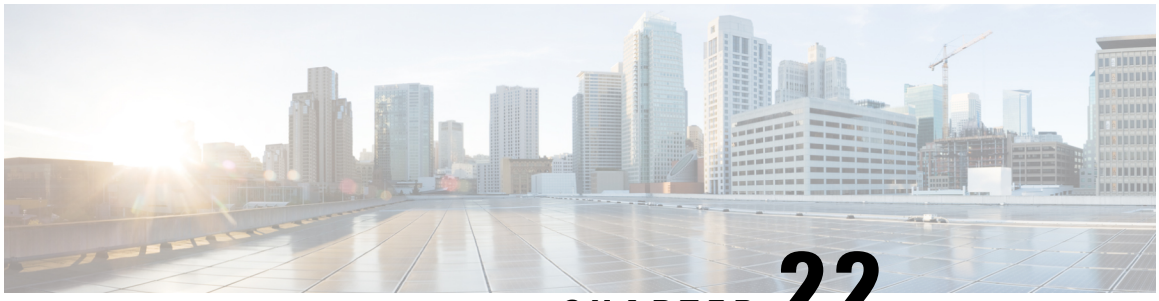
CVP Graceful Shutdown

To perform graceful shutdown of the CVP Call Server and the associated Reporting Server, use the gracefulShutdown CVP API. The API initiates the PUT operation by specifying the device type for Call Server and IP address & device type for the Reporting Server.

The gracefulShutdown is invoked on the following specified components:

- CVP_CALL_SERVER
  - SubsystemIVR
  - SubsystemSIP
  - SubsystemGED
  - SubsystemMessageBus
- CVP_RPT_SERVER
  - SubsystemReporting

For more information on the REST operation, refer the **CVP Graceful Shutdown API** section in the *Unified CVP API Developer Guide* at https://developer.cisco.com/site/customer-voice-portal/documents/rest-api/.



CHAPTER 22

Data Center API

This API allows administrators to manage remote data center in the Packaged CCE machine inventory. This is for Packaged CCE 2000 Agents deployment. For information on Data Center API for 4000 Agents and 12000 Agents deployment, see Inventory Import API, on page 123.

Note The term "data center" refers to site (remote site/main site) in PCCE deployment.

URL

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

Operations

- create: Creates a new data center and stores it in the database.
- delete: Deletes one data center from the database.
  `https://<server>/unifiedconfig/config/datacenter/{id}`
  - This operation deletes all the data center configuration only if there are no agents, teams, dialed numbers, route patterns, or skill groups associated to the data center. Otherwise it returns an error.
  - This operation deletes the external machines configured to the remote data center.
- get: Returns one data center, using the URL
  `https://<server>/unifiedconfig/config/datacenter/<id>`
- list: Retrieves a list of data centers configured in the database.
- update: Updates one data center using the URL
  `https://<server>/unifiedconfig/config/datacenter/<id>`
  - You cannot change the data center name, and Side A and Side B PG addresses.
  - You can add new PG types but cannot remove the existing ones.

Parameters

- name: Required. The unique name of the data center. Maximum length is ten characters. Valid characters are alphanumeric, period (.), and underscore (_). The first character must be alphanumeric.
- agentPG: Information about the agent PG for the remote data center, which includes the following:
  - configured: Indicates whether an agent PG is configured for the remote data center. True or false. If the value is false, the rest of the values for the agent PG are ignored.
  - cmSubSideA:
    - name: Name of the Call Manager subscriber sideA. Maximum length is ten characters. Valid characters are alphanumeric, period (.), and underscore (_). The first character must be alphanumeric.
    - refURL: Reference to Call Manager subscriber sideA.
  - cmSubSideB:
    - name: Name of the Call Manager subscriber sideB. Maximum length is ten characters. Valid characters are alphanumeric, period (.), and underscore (_). The first character must be alphanumeric.
    - refURL: Reference to Call Manager subscriber sideB.
  - finessePrimaryAddress: Address of Finesse primary.
  - finesseUserName: Username to access Finesse primary.
  - finessePassword: Password to access Finesse primary.
- vruPG: Information about the VRU PG for the remote data center, which includes the following:
  - configured: Indicates whether a VRU PG is configured for the remote data center. True or false. If the value is false, the rest of the values for the VRU PG are ignored.
  - cvp1Address: Address of the CVP1 server.
  - cvp2Address: Address of the CVP2 server.
- mrPG: Information about the MR PG for the remote data center, which includes the following:
  - configured: Indicates whether an MR PG is configured for the remote data center. True or false. If the value is false, the rest of the parameters for the MR PG are ignored.

Search and Sort Parameters

The following table shows the parameters that are searched and the parameters that are sortable.

Search Parameters	Sort Parameters
name	name

External Machines

Remote data centers support the following types of external machines:

- CVP Reporting Server
- Gateways
- Cisco Virtualized Voice Browser (CVVB)
- Cisco Unified SIP Proxy (CUSP) Server
- Enterprise Chat and Email
- SocialMiner
- Third Party Multichannel

It allows you to create, update, and delete the external machines. You can create only one external machine of each type in a remote data center. You can add the following external machines based on the PG client type configured in the remote data center

- Agent: None
- VRU: Unified CVP Reporting
- Multichannel: Third-party Multichannel, Enterprise Chat and Email, and SocialMiner

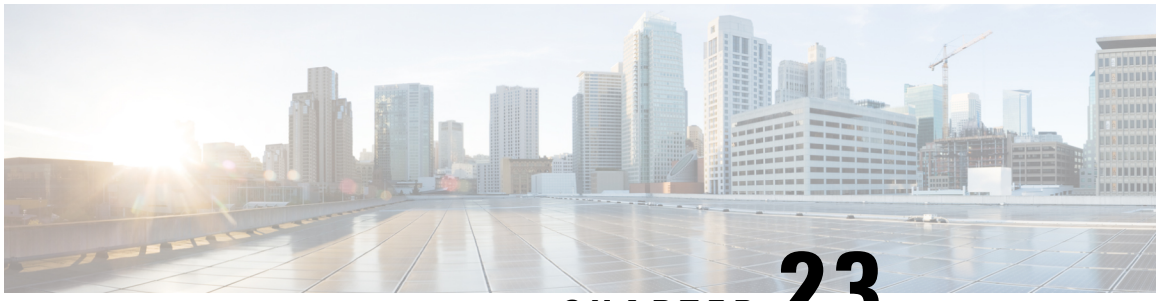
When you add an external CVP reporting server to a remote data center, the CVP call server of that data center associates to this reporting server.

When you delete the external CVP reporting server, the call servers of the remote data center re-associates to the core data center CVP reporting server.

Example Get Response

```
<datacenter>
    <name>Boston</name>
    <refURL>/unifiedconfig/config/datacenter/5001</refURL>
    <changeStamp>1</changeStamp>
    <sideAPGHostName>pg1a.boston.icm</sideAPGHostName>
    <sideBPGHostName>pg1b.boston.icm</sideBPGHostName>
 <agentPG>
        <configured>true</configured>
        <cmSubSideA>
               <refURL>/unifiedconfig/config/machineinventory/1234</refURL>
               <name>cmSubA.boston.icm</name>
        </cmSubSideA>
        <cmSubSideB>
               <refURL>/unifiedconfig/config/machineinventory/5678</refURL>
               <name>cmSubB.boston.icm</name>
        </cmSubSideB>
        <finessePrimaryHostName>finesseP.boston.icm</finessePrimaryHostName>
        <finesseUserName>boston</finesseUserName>
    </agentPG>
    <vruPG>
        <configured>true</configured>
        <cvp1HostName>cvp1.boston.icm</cvp1HostName>
        <cvp2HostName>cvp2.boston.icm</cvp2HostHame>
    </vruPG>
    <mrPG>
```

```
        <configured>true</configured>
    </mrPG>
</datacenter>
```



CHAPTER 23

Department API

Packaged CCE allows you to create departments, add configuration items to departments, and assign administrators to departments to limit the scope of their control. For example, the call center for a hospital might have departments for Radiology, Surgery, and Cardiology. Use of departments is optional.

For more information on how departments work, see the *Cisco Packaged Contact Center Enterprise Administration and Configuration Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html.

Use the Department API to list the departments currently defined in the database, define new departments, and view, edit, and delete existing departments.

URL

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

Operations

- create: Creates one department.
- delete: Marks one department for deletion.
- get: Returns one department, using the URL `https://<server>/unifiedconfig/config/department/<id>`.
- list: Retrieves a list of departments.
- update: Updates one department.

Note When you create, update or delete a department in Packaged CCE, the corresponding operations takes place in the Enterprise Chat and Email as well.

Parameters

- refURL: The refURL of the department. See Shared Parameters, on page 8.
- name: The name of this department. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.

- description: See Shared Parameters, on page 8.
- administrators: A collection of administrator (Administrator API, on page 15) references associated with this department, including the refURL, user name, and domain name. See References, on page 5.

Search and Sort Values

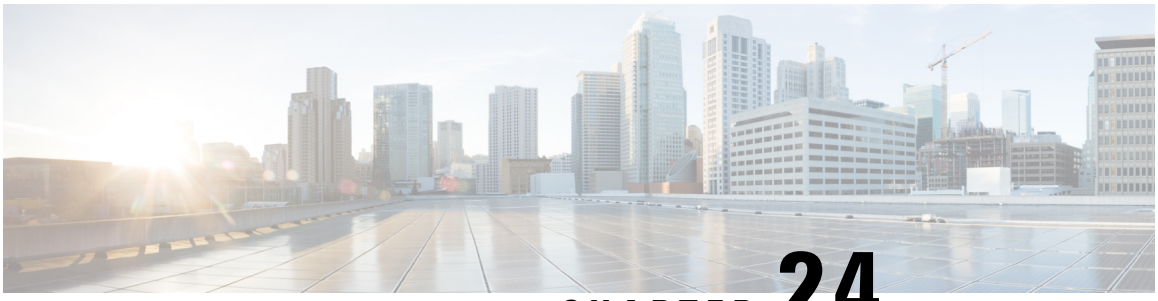
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<department>
   <changeStamp>0</changeStamp>
   <refURL>/unifiedconfig/config/department/(id)</refURL>
   <name>department1</name>
   <description>test department1</description>
   <administrators>
      <administrator>
         <refURL>/unifiedconfig/config/administrator/(id_1)</refURL>
         <userName>JohnSmith</userName>
         <domainName>BOSTON.COM</domainName>
      </administrator>
      <administrator>
         <refURL>/unifiedconfig/config/administrator/(id_2)</refURL>
         <userName>JaneDoe</userName>
         <domainName>BOSTON.COM</domainName>
      </administrator>
   </administrators>
</department>
```



CHAPTER 24

Deployment API

The Deployment API is used to view the deployment type of the installation. It is read-only, and does not require authentication. To change the deployment type, use the Deployment Type Info API.

URL

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

Parameters

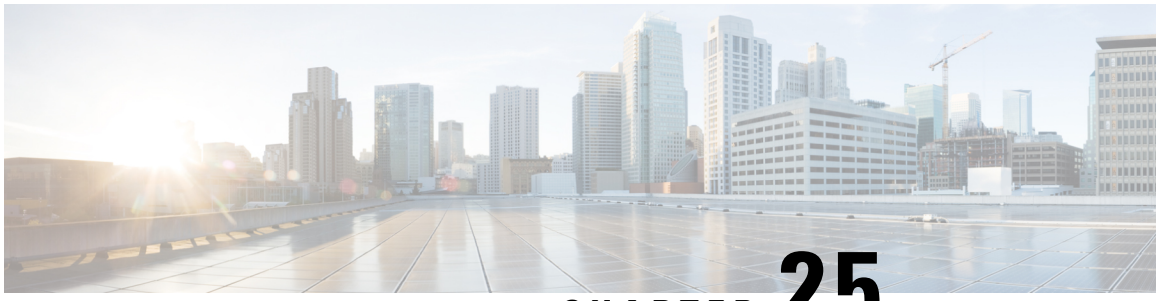
- deploymentType: The type of deployment. See Deployment Type Info API, on page 93.
- supervisorLoginAllowed: Indicates whether the current deployment type allows supervisor login.

Operations

- get: Returns the deployment type of the installation using the URL `https://<server>/unifiedconfig/config/deployment`.

Example Get Response

```
<deployment>
     <deploymentType>7</deploymentType>
     <supervisorLoginAllowed>true</supervisorLoginAllowed>
</deployment>
```



CHAPTER 25

Deployment Type Info API

Use the Deployment Type Info API to view or edit the current system deployment type.

URL

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

Operations

- get: Returns the current deployment type and the results of the capacity and system validation tests, using the URL `https://<server>/unifiedconfig/config/deploymenttypeinfo`.
- update: Sets the specified deployment type if the system validation check, capacity check, and VM Validation for that deployment type pass and are required.

Parameters

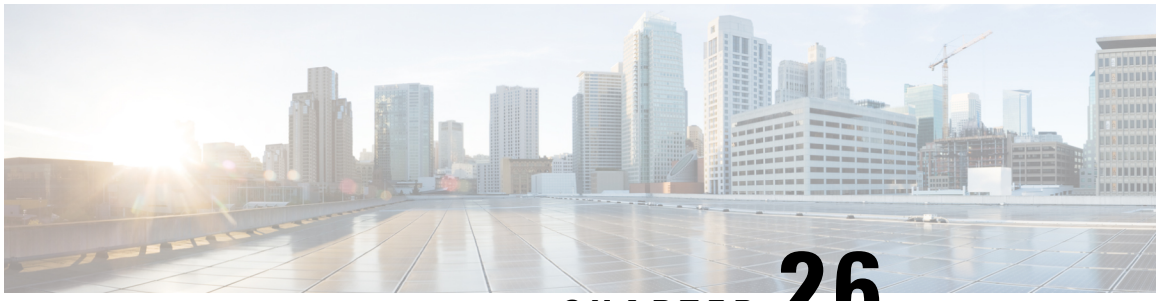
- changeStamp: See Shared Parameters, on page 8.
- vmHosts: vmHost information, including name, address, username, and password parameters of Side A and Side B. Only required when switching to Packaged CCE, to allow access to the ESX servers for VM validation.
- permissionInfo: See Permissions, on page 9.
- systemValidationStatus: See Serviceability API, on page 199.
- capacityInfo: See Serviceability API, on page 199.
- vmLayoutType: Represents the current deployment type and hardware.
- vmValidationLogURL: The URL to download a file about VM layout validation.
- deploymentType: The type of deployment. The following types are supported:
  - 0: No deployment type specified. Initial type set at installation. Once set to another deployment type, you cannot switch back to 0.
  - 1: NAM (Deprecated)
  - 2: Contact Director
  - 3: NAM Rogger (Deprecated)

- 4: ICM Router/Logger
- 5: UCCE: 8000 Agents Router/Logger
- 6: UCCE: 12000 Agents Router/Logger
- 7: Packaged CCE: 2000 Agents
- 8: ICM Rogger
- 9: UCCE: 4000 Agents Rogger
- 10: Packaged CCE: Lab Mode
- 11: HCS-CC: 2000 Agents
- 13: UCCE: Progger (Lab Only)
- 14: HCS-CC: 4000 Agents
- 15: HCS-CC: 12000 Agents
- 16: UCCE: 2000 Agents
- 17: Packaged CCE: 4000 Agents
- 18: Packaged CCE: 12000 Agents
- 19: UCCE: 24000 Agents Router/Logger
- 20: HCS-CC: 24000 Agents

- targetDeploymentType: Indicates which deployment type is being initialized.
- hardwareLayoutType: Indicates the hardware layout types - Tested Reference Configuration (TRC) or Specification based hardware (SPEC). This is an optional parameter. If the hardware type is not specified, the system takes TRC by default.

Example Get Response

```
<deploymentTypeInfo>
  <changeStamp>59</changeStamp>
  <deploymentType>7</deploymentType>
  <vmLayoutType>PCCE_C240M3_Full</vmLayoutType>
  <vmHosts>
    <vmHost>
      <name>sideA</name>
      <address>10.86.141.10</address>
      <userName>root</userName>
    </vmHost>
    <vmHost>
      <name>sideB</name>
      <address>10.86.141.29</address>
      <userName>root</userName>
      <password>pwexample</password>
    </vmHost>
  </vmHosts>
</deploymentTypeInfo>
```



CHAPTER 26

Dialed Number API

Dialed numbers are string values used to select the appropriate routing script so that a voice call or a non-voice task (such as an email or a request for a web chat) can be delivered to an agent.

Use the Dialed Number API to list the dialed numbers currently defined in the database, define new dialed numbers, and view, edit, and delete existing dialed numbers.

URL

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

Operations

- create: Creates one dialed number.
- delete: Marks one dialed number for deletion.
- get: Returns one dialed number, using the URL `https://<server>/unifiedconfig/config/dialednumber/<id>`.
- list: Retrieves a list of dialed numbers.
  - **Query parameters:**
    - Summary list: See list, on page 3.
- update: Updates one dialed number.

Parameters

- dialedNumberString: Required. Value used to route the call or direct the non-voice task. A unique string for the routing type. Maximum of 25 characters. The External Voice and Post Call Survey routing types cannot have the same dialed number string.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- department: A reference to the department (Department API, on page 89), including the refURL and name. See References, on page 5.
- routingType: Specifies where a call or non-voice task request originates.

Refer to the Routing Type API, on page 191 to determine the routing types on which you can create a dialed number.

- External Voice. Calls come from Unified CVP. When creating a Dialed Number using this type, a dialed number database record is created for Unified CVP routing client.
- Post Call Survey. Calls come from Unified CVP. When creating a Post Call Survey Dialed Number using this type, a dialed number database record is created for Unified CVP routing client.
- Internal Voice. Calls come from a Unified CM phone.
- Outbound. Calls that come from the Outbound Option Dialer.
- Multichannel 1. Requests that come from an Enterprise Chat and Email, SocialMiner, or third party.
- Multichannel 2. Requests that come from an Enterprise Chat and Email, SocialMiner, or third party.
- Multichannel 3. Requests that come from an Enterprise Chat and Email, SocialMiner, or third party.

- callType: A reference to a call type (Call Type API, on page 61) for this dialed number, including a refURL and name. See References, on page 5.
- dialedNumberRecords: A collection of dialed number record entries each containing the id and name of a dialed number database record. Read-only.
- mediaRoutingDomain: A reference to the media routing domain (Media Routing Domain API, on page 143) for the dialed number. See References, on page 5.
- peripheralSet: A reference to a peripheral set for the dialed number.This parameter is mandatory for Packaged CCE 4000 Agents and 12000 Agents deployment type.

  The peripheralSet parameter is not available for Packaged CCE 2000 Agents deployment type.
- datacenter: A reference to the data center, including the refURL and name.

  This parameter is mandatory for Packaged CCE 4000 Agents or 12000 Agents deployment type. You must provide the reference to a data center that contains above peripheral set. For more information on data center for 4000 Agents or 12000 Agents deployment, see Inventory Import API, on page 123.
- pcsEnabledDialedNumberPattern: Optional. Dialed Numbers (DN) or DN patterns to associate with the Post Call Survey number. The maximum character length is 512 that can contain comma separated list without any spaces. Applicable only when the Routing Type is 7 (Post Call Survey -Unified CVP). If the Routing Type is not 7, the system returns an error message.
- ringtoneName: Optional. A file name that has the customized ringtone. The file name can have a maximum length of 256 characters. Internationalized characters are allowed. Applicable only when the Routing Type is 1 (External Voice - Unified CVP). If the Routing Type is not 1, the system returns an error message.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• dialedNumberString • description	• dialedNumberString (default) • description • datacenter.name • pcsEnabledDialedNumberPattern • peripheralSet.name (Available for Packaged CCE 4000 Agents and 12000 Agents deployment type.)

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

The Dialed Number API also supports advanced search parameters, such as routing type (Routing Type API, on page 191) and data center (Data Center API, on page 85).

- **routingType:<type>** Finds all dialed numbers with the specified routing type value. Valid types match those in the routingType parameter.
  - **routingType:1** Returns all dialed numbers with an external voice routing type.
- **datacenters:(dc1|dc2|dc3...)** which returns all dialed numbers which belong to any of the specified data centers. You can specify up to three data centers. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all dialed numbers in the core data center.
- **peripheralsets: (ps1|ps2|ps3...)** returns the dialed numbers specific to the peripheral sets in 4000 Agents and 12000 Agents deployment. The peripheral set names are fully matched (case-insensitive, no partial matches).

Example Get Response

```
 <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>
  <datacenter>
    <name>Boxborough</name>
    <refURL>/unifiedconfig/config/datacenter/5000</refURL>
  </datacenter>
    <callType>
    <refURL>/unifiedconfig/config/calltype/(id)</refURL>
    <name>calltype name</name>
  </callType>
  <dialedNumberRecords>
    <dialedNumberRecord>
      <id>10</id>
      <name>C5011.8885551212</name>
    </dialedNumberRecord>
```

```
    <dialedNumberRecord>
      <id>11</id>
      <name>C5012.8885551212</name>
    </dialedNumberRecord>

  </dialedNumberRecords>
<pcsEnabledDialedNumberPattern>1800*,1700*</pcsEnabledDialedNumberPattern>
<ringtoneName>abc.wav</ringtoneName>
<routingType>7</routingType>
</dialedNumber>
```

Following are the REST responses received during execution of REST API to configure the dialed number:

- **Success** - Configuration changes persist in AW DB and synchronized with respective devices.

  Code: 200

  Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with one or more devices.

  Code: 201

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with CVP failed.)

  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

  ```
  <apiErrors>
  <apiError>
  <errorMessage>Configuration update failed for one or more devices.</errorMessage>
  <errorType>PARTIAL_SUCCESS</errorType>
  </apiError>
  </apiErrors>
  ```

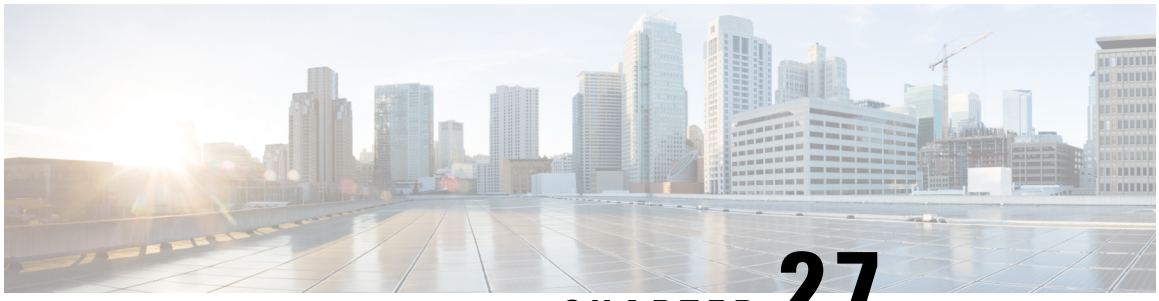
  Note The partial success generally occurs for dialed number configured with Post Call Survey (PCS) and/or Ringtone.

- **Failure**- The configuration updates to AW DB is failed.

Example Get Response for Packaged CCE 4000 Agents and 12000 Agents Deployment

```
<dialedNumber>

<refURL>[https://***.***.***.***/unifiedconfig/config/dialedNumber/(id)]</refURL>
  <description>test dialed number</description>
  <dialedNumberString>8885551212</dialedNumberString>
  <routingType>1</routingType>
  <changeStamp>0</changeStamp>
  <mediaRoutingDomain>
    <refURL>[https://***.***.***.***/unifiedconfig/config/mediaroutingdomain/1]<refURL>
    <name>Cisco_Voice</name>
  </mediaRoutingDomain>
```

```
  <peripheralSet>
    <refURL>/unifiedconfig/config/inventory/datacenter/(datacenter
name)/peripheralset/(id)</refURL>
    <name>rack1</name>
  </peripheralSet>
  <callType>
    <refURL>[https://***.***.***.***/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>
  </dialedNumberRecords>
 </dialedNumber>
```



CHAPTER 27

ECC Payload API

You use expanded call variables to store values that are associated with a call. These variables are commonly called Expanded Call Context (ECC) variables. While you can define many more ECC variables in the system, you can pass a maximum of 2000 bytes of these variables at any time over any interface. To enable you to pass different ECC variables at different times, the solution has ECC payloads.

An ECC payload is a defined set of ECC variables. You can create ECC payloads to suit the necessary information for a given operation. You can include an ECC variable in multiple ECC payloads. The particular ECC variables in a given ECC payload are called its members.

Note

- ECC payload ID 1 is the default ECC payload. The API prevents the deletion of this ECC payload.
- In solutions that only use the default ECC payload, the Logger does not create an ECC variable that exceeds the 2000-byte limit for an ECC payload or the 2500-byte CTI Message Size limit. The Logger does this because it automatically adds all ECC variables to the default ECC payload if that is the only ECC payload.

  If you create another ECC payload, the Logger no longer checks the 2000-byte limit when creating ECC variables. The Logger creates the ECC variables without assigning them to an ECC payload. Assign the new ECC variable to an appropriate ECC payload yourself, either through the ECC Payload API or through the ECC Payload Tool.
- For create and update operations, the ECC Payload API verifies that the members of the ECC payload do not exceed the 2000-byte limit.

Important

During upgrades, when the system first migrates your existing ECC variables to the Default payload, it does not check the CTI message size limit. The member names might exceed the extra 500 bytes that is allocated for ECC payloads to a CTI client. Manually check the **CTI Message Size** counter in the **Expanded Call Variable Payload List** tool to ensure that the Default payload does not exceed the limit. If the Default payload exceeds the limit, modify it to meet the limit.

URL

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

Operations

- create: Creates one ECC payload and stores it in the database.
  - **Query Parameters**
    - name: The name of the ECC payload (Required)
    - description: A description of the ECC payload (Optional)
    - variables: 0 or more ECC variables (Optional)

Note Specify each variable with the `ecc.refURL` of a valid, non-deleted ECC variable.

Note This API supports synchronous and asynchronous create operations.

- delete: Permanently deletes one ECC payload and all its members from the database.
- get: Returns one ECC payload from the database, using the URL `https://<server>/unifiedconfig/config/eccpayload/<id>`.
- list: Retrieves a list of ECC payloads.
- update: Updates one ECC payload in the database.
  - **Query Parameters**
    - changeStamp: The change stamp for the ECC payload record which the GET returns (Required)
    - refURL: The `refURL` of the ECC payload to update (Required)
    - name: The name of the ECC payload (Optional)
    - description: A description of the ECC payload (Optional)
    - variables: 0 or more ECC variables (Optional)

Note Specify each variable with the `ecc.refURL` of a valid, non-deleted ECC variable.

Parameters

- refURL: The refURL of the ECC payload. See Shared Parameters, on page 8.
- name: The name of the ECC payload. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- variables: The members of the ECC payload.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

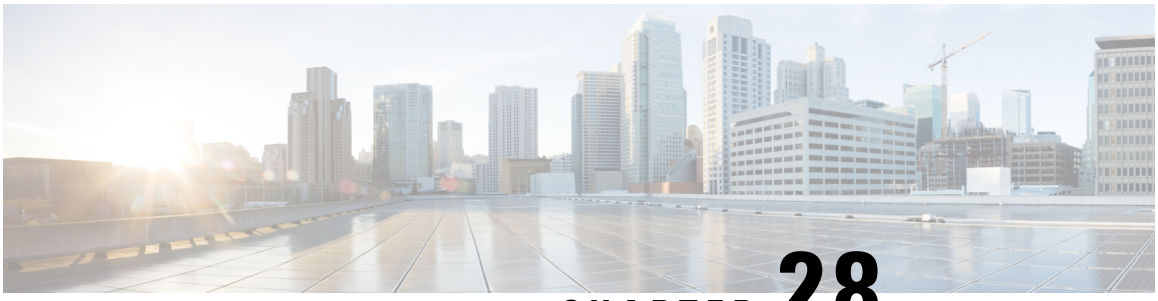
Search parameters	Sort parameters
• name • description	• name (default) • description

See Search, on page 10 and Sort, on page 11.

Note This API does not support advanced search parameters.

Example Get Response

```
<eccpayload>
  <refURL>/unifiedconfig/config/eccpayload/1</refURL>
  <changeStamp>18</changeStamp>
  <description>Initial default data.</description>
  <name>Default</name>
  <variables>
    <variable>
      <refURL>/unifiedconfig/config/expandedcallvariable/8</refURL>
      <name>POD.ID</name>
    </variable>
    <variable>
      <refURL>/unifiedconfig/config/expandedcallvariable/5009</refURL>
      <name>user.A1</name>
    </variable>
    <variable>
      <refURL>/unifiedconfig/config/expandedcallvariable/5010</refURL>
      <name>user.A2</name>
    </variable>
  </variables>
</eccpayload>
```



CHAPTER 28

Expanded Call Variable API

Calls carry data with them as they move through the system. This data, called expanded call variable data, is embedded with the call and is visible on the agent desktop.

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, and delete existing expanded call variables.

URL

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

Operations

- create: Creates one expanded call variable.

Note In solutions that only use the default ECC payload, the Logger does not create an ECC variable that exceeds the 2000-byte limit for an ECC payload. The Logger does this because it automatically adds all ECC variables to the default ECC payload if that is the only ECC payload.

If you create another ECC payload, the Logger no longer checks the 2000-byte limit. The Logger creates ECC variables without automatically assigning them to an ECC payload. Assign the new ECC variable to an appropriate ECC payload yourself, either through the ECC Payload API or through the ECC Payload Tool.

- delete: Marks one expanded call variable for deletion, but does not permanently delete it.
- get: Returns one expanded call variable, using the URL `https://<server>/unifiedconfig/config/expandedcallvariable/<id>`.
- list: Retrieves a list of expanded call variables.
- update: Updates one expanded call variable.

Parameters

- refURL: The refURL of the expanded call variable. See Shared Parameters, on page 8.
- name: The name of the expanded call variable. See Shared Parameters, on page 8.

- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- maximumLength: The maximum length of the expanded call variable. The value is 1 to 210.
- eccArray: Indicates whether the expanded call variable is an array. Values are true/false.
- maximumArraySize: The maximum number of elements in the array is 1 to 255. Required if eccArray is true; must be blank or not specified if eccArray is false.
- enabled: Indicates whether the expanded call variable is enabled. Values are true/false.
- persistent: Specifies whether the expanded call variable is written to the historical database with each Termination Call Detail and Route Call Detail record. Values are true/false.
  - No persistent, enabled arrays are allowed.
  - The maximum number of persistent, scalar, enabled variables is 20.
- ciscoProvided: Indicates whether the expanded call variable is provided by Cisco. Values are true/false. Read-only.
- bytesRequired: The number of bytes required to store the expanded call variable in the system. Read-only.

  The size 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.
- bytesRequiredInCtiServer: The number of bytes required to send this variable to CTI Server. Read-only.

  The size 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.
  - The total bytesRequired of all the ECC variables in an ECC payload cannot exceed 2000 bytes.
  - The total bytesRequiredInCtiServer of all the ECC variables in an ECC payload cannot exceed 2500 bytes (2000 bytes for the values and 500 bytes for the ECC variable names).

Search and Sort Values

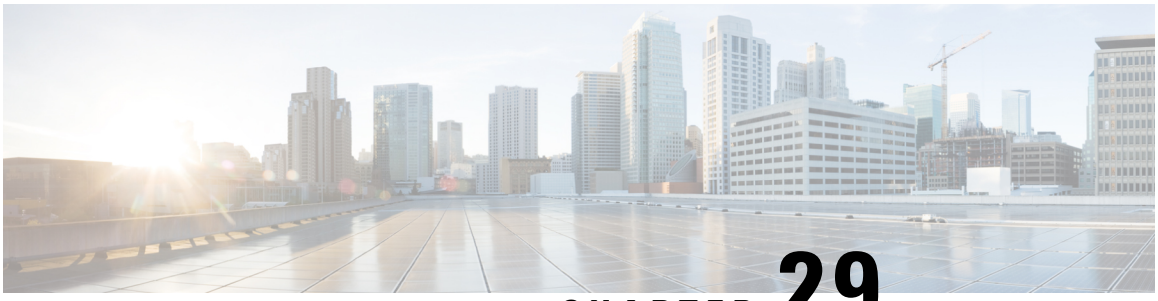
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description • maximumLength • maximumArraySize • eccArray • enabled • persistent • ciscoProvided

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<expandedCallVariable>
    <refURL>/unifiedconfig/config/expandedcallvariable/(id)</refURL>
    <name>test</name>
    <maximumLength>9</maximumLength>
    <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>
```



CHAPTER 29

Facility API

Use the Facility API to retrieve a list of every facility, including associated instances, from the Active Directory domain. This API is read-only.

URL

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

Operations

- list: Returns a list of facilities.

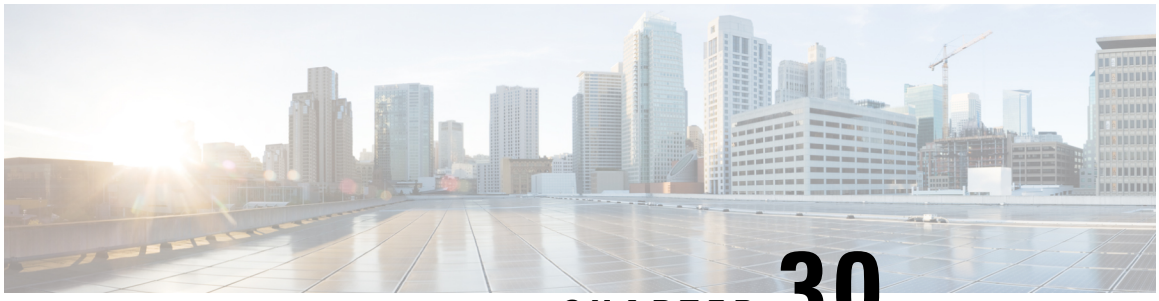
Parameters

- facilities: A collection of facility items, including a list of each facility's instances. Includes a name parameter.

Example Get Response

```
<results>
   <facilities>
      <facility>
         <instances>
            <instance>
               <name>bos01</name>
            </instance>
            <instance>
               <name>pra01</name>
            </instance>
            <instance>
               <name>bos02</name>
            </instance>
         </instances>
         <name>bos</name>
      </facility>
      <facility>
         <instances>
            <instance>
               <name>test</name>
            </instance>
         </instances>
         <name>Cisco_test_fac</name>
      </facility>
```

```
   </facilities>
</results>
```



CHAPTER 30

Global API

The Global API returns global settings from the following categories:

- Call
- Agent
- Reporting
- Script
- Labels

Note Labels are applicable for Packaged CCE 2000 agent deployment only.

URL

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

Operations

- list: Retrieves a list of global settings.
- update: Updates global settings.

Parameters

callReporting

- defaultBucketInterval: Required. A reference to a bucket interval (Bucket Interval API, on page 47), including refURL and name. See References, on page 5.
- defaultCallType: Required. A reference to a call type (Call Type API, on page 61), including refURL and name. A call is categorized against this call type unless it comes into the system on a dialed number that is associated with another call type. See References, on page 5.
- serviceLevelType: Required. This value indicates how the system calculates the service level.
  - 1: Ignore Abandoned Calls.

- 2: Abandoned Calls have Negative Impact.
- 3: Abandoned Calls have Positive Impact.

- serviceLevelThreshold: Required. Maximum time in seconds that a caller should wait before being connected with an agent. Maximum is 86,400 seconds (1 day).
- abandonCallWaitTime: Required. Configures the minimum time an incoming call must be queued before the call is considered abandoned if the caller hangs up. Maximum is 14400 seconds (4 hours).
- answeredShortCallThreshold: Configures the maximum duration for a short call. Calls with a duration below that value are considered short. Value is between 0 and 14400 seconds (4 hours).

agent

- agentPhoneLineControl: Indicates whether all agents supported on the agent peripheral can have one or more than one line configured.
  - 0: Single Line.
  - 1: All Lines.
- nonACDLineImpact: Specifies how the agent state is set when the agent is on a call on a secondary line and agentPhoneLineControl is set to All Lines.
  - 0: Available agent stays available.
  - 1: Available agent goes not ready.
- defaultDeskSetting: A reference to a desk setting (Agent Desk Settings API, on page 25), including refURL and name.
- loginNameCaseSensitivity: Identifies whether usernames are case-sensitive. Values are true/false.
- minimumPasswordLength: Changing this value affects new passwords only and does not apply to existing ones. Value is between 0 and 32.

reporting

- reportingInterval: Configures the system to store historical information in 15-minute or 30-minute summaries. The 15-minute interval requires a larger amount of database space than the 30-minute interval. Values are 15 or 30.

script

- retainScriptVersion: Defines the maximum number of versions of each routing script to maintain in the database. The system automatically deletes the oldest version when the limit is exceeded. Maximum is 100.

labels

- cmLabel: Pattern that matches the Unified CM route pattern. Must be a 10-digit string.
- cvpLabels: A collection of labels that include the pattern and routingClientName. The patterns should match the CVP Dialed Number patterns. Must be a 10-digit string.
- outboundLabel: Pattern that matches IOS Voice Gateway dial-peer. Must be a 10-digit string.

datacenterSettings

- A collection of data center settings including Agent and Labels categories. The refURL field is required for each data center setting. Under Agent, you can update the following parameters if Agent PG is created for this data center:
  - agentPhoneLineControl
  - nonACDLineImpact
  - defaultDeskSetting

  Under Labels, you can update certain or all labels depending on the type of PGs that exist in the data center. For example, if only Agent PG is configured in the data center, you can update only cmLabel.

Example Get Request

```
<globalSettings>
        <callReporting>
            <serviceLevelType/>
            <serviceLevelThreshold/>
            <abandonCallWaitTime/>
            <answeredShortCallThreshold/>
            <defaultCallType>
                <refURL/>
                <name/>
            </defaultCallType>
            <defaultBucketInterval>
                <refURL/>
                <name/>
            </defaultBucketInterval>
        </callReporting>
        <agent>
           <nonACDLineImpact/>
            <agentPhoneLineControl/>
            <defaultDeskSetting>
                <refURL>/unifiedconfig/config/agentdesksetting/5000</refURL>
                <name>Default_Agent_Desk_Settings</name>
            <defaultDeskSetting/>
            <loginNameCaseSensitivity/>
            <minimumPasswordLength/>
        </agent>
        <reporting>
            <reportingInterval/>
        </reporting>
        <script>
            <retainScriptVersion/>
        </script>
        <labels>
            <cmLabel>8881111000</cmLabel>
            <outboundLabel>6661111000</outboundLabel>
            <cvpLabels>
                <cvpLabel>
                    <routingClientName/>
                    <pattern/>
             </cvpLabel>
        </labels>
        <datacenterSettings>
            <datacenterSetting>
            <datacenter>
                <refURL> /unifiedconfig/config/datacenter/5000</refURL>
                <name>boston</name>
```

```
            </datacenter>
            <agent>
                <nonACDLineImpact/>
                <agentPhoneLineControl/>
                <defaultDeskSetting>
                    <refURL>/unifiedconfig/config/agentdesksetting/5000</refURL>
                    <name>Default_Agent_Desk_Settings</name>
                <defaultDeskSetting/>
            <agent>
            <labels>
                <cmLabel/>
                <outboundLabel/>
                <cvpLabels>
                    <cvpLabel>
                        <routingClientName/>
                        <pattern/>
                    </cvpLabel>
                </cvpLabels>
            </labels>
        </datacenterSetting>
    </datacenterSettings>
</globalSettings>
```

Example Update Request

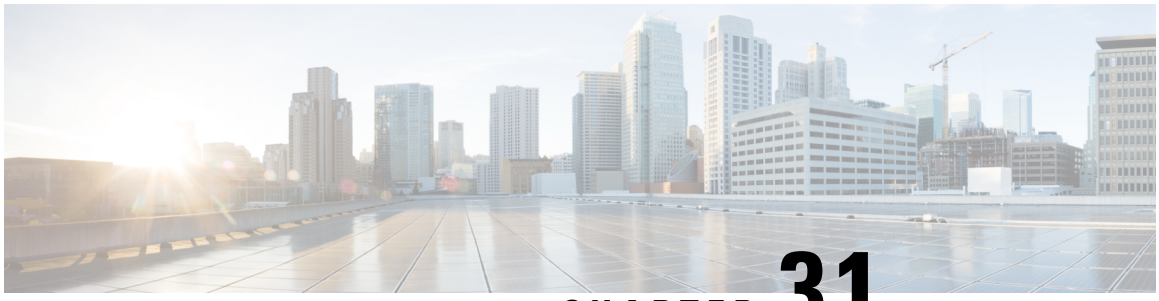
```
<globalSettings>
              <changeStamp>59</changeStamp>
              <callReporting>
                     <serviceLevelType>2</serviceLevelType>
                     <serviceLevelThreshold>10</serviceLevelThreshold>
                     <abandonCallWaitTime>300</abandonCallWaitTime>
                     <answeredShortCallThreshold>30</answeredShortCallThreshold>
                     <defaultCallType>
                            <refURL>/unifiedconfig/config/calltype/5000</refURL>
                     </defaultCallType>
                     <defaultBucketInterval>
                            <refURL>/unifiedconfig/config/bucketinterval/5001</refURL>
                     </defaultBucketInterval>
              </callReporting>
              <agent>
                    <agentPhoneLineControl>1</agentPhoneLineControl>
                    <nonACDLineImpact>0</nonACDLineImpact>
                    <defaultDeskSetting>
                         <refURL>/unifiedconfig/config/agentdesksetting/5003</refURL>
                    </defaultDeskSetting>
                   <loginNameCaseSensitivity>true</loginNameCaseSensitivity>
                    <minimumPasswordLength>8</minimumPasswordLength>
              </agent>
              <reporting>
                     <reportingInterval>15</reportingInterval>
              </reporting>
              <script>
                     <retainScriptVersion>5</retainScriptVersion>
              </script>
              <labels>
                      <cmLabel>8881111000</cmLabel>
                      <outboundLabel>6661111000</outboundLabel>
                      <cvpLabels>
                           <cvpLabel>
                               <routingClientName>CVP_PG_1A</routingClientName>
                               <pattern>7777777771</pattern>
                           </cvpLabel>
                           <cvpLabel>
                               <routingClientName>CVP_PG_1B</routingClientName>
```

```
                        <pattern>7777777772</pattern>
                    </cvpLabel>
                </cvpLabels>
            </labels>
            <datacenterSettings>
               <datacenterSetting>
                   <datacenter>
                          <refURL> /unifiedconfig/config/datacenter/5000</refURL>
                   </datacenter>
                   <agent>
                          <agentPhoneLineControl>1</agentPhoneLineControl>
                          <nonACDLineImpact>0</nonACDLineImpact>
                          <defaultDeskSetting>
            <refURL>/unifiedconfig/config/agentdesksetting/5003</refURL>
                                       </defaultDeskSetting>
            </agent>
            <labels>
                   <cmLabel>8881111000</cmLabel>
                   <outboundLabel>6661111000</outboundLabel>
                   <cvpLabels>
                   <cvpLabel>
            <routingClientName>boston_CVP_PG_1A</routingClientName>
                                       <pattern>7777777771</pattern>
                               </cvpLabel>
                               <cvpLabel>
            <routingClientName>boston_CVP_PG_1B</routingClientName>
                                       <pattern>7777777772</pattern>
                               </cvpLabel>
                        </cvpLabels>
                   </labels>
            </datacenterSetting>
      </datacenterSettings>
 </globalSettings>
```

Example Get/Update Request for Packaged CCE 4000 Agents and 12000 Agents Deployment

```
<globalSettings>
<changeStamp>59</changeStamp>
<callReporting>
<serviceLevelType>2</serviceLevelType>
<serviceLevelThreshold>10</serviceLevelThreshold>
<abandonCallWaitTime>300</abandonCallWaitTime>
<answeredShortCallThreshold>30</answeredShortCallThreshold>
<defaultCallType>
<refURL>/unifiedconfig/config/calltype/5000</refURL>
</defaultCallType>
<defaultBucketInterval>
<refURL>/unifiedconfig/config/bucketinterval/5001</refURL>
</defaultBucketInterval>
</callReporting>
<agent>
<agentPhoneLineControl>1</agentPhoneLineControl>
<nonACDLineImpact>0</nonACDLineImpact>
<defaultDeskSetting>
<refURL>/unifiedconfig/config/agentdesksetting/5003</refURL>
</defaultDeskSetting>
<loginNameCaseSensitivity>true</loginNameCaseSensitivity>
<minimumPasswordLength>8</minimumPasswordLength>
</agent>
<reporting>
<reportingInterval>15</reportingInterval>
</reporting>
<script>
<retainScriptVersion>5</retainScriptVersion>
```

```
</script>
</globalSettings>
```



CHAPTER 31

Initialize API

The initialize API serves as an entry point for Packaged CCE system setup. It allows you to:

- Start setup tasks for system configuration.
- Check the initialization status of the system.

URL

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

Operations

- list: Lists information about the system initialization status.
- update: Starts a system initialization.

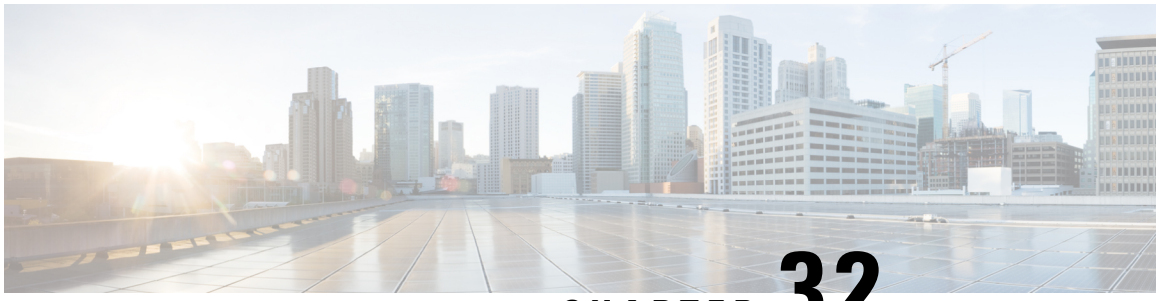
Parameters

- name: The name of the startup task.
- state: The state of the task. Values are:
  - NOT_STARTED
  - PROCESSING
  - FAILED_NEEDS_RETRY: Occurs when an initialization task that does not require uninitialization fails. Correct the errors then invoke the Initialize API again.
  - FAILED: Occurs when a task that requires uninitialization fails. Correct the errors, and then invoke the Uninitialize API. After a successful uninitialization, the initialization request can be made again.
  - SUCCEEDED

Example Get Response

```
<results>
   <state>FAILED</state>
   <initializationStatuses>
      <initializationStatus>
         <state>NOT_STARTED</state>
         <name>JtapiClientTask</name>
```

```
      </initializationStatus>
      <initializationStatus>
         <state>FAILED</state>
         <name>PGUserTask</name>
         <apiErrors>
             <apiError>
                 <errorType>errorConnectAXL</errorType>
                 <errorMessage>There is no UCM AXL service defined yet</errorMessage>
             </apiError>
          </apiErrors>
      </initializationStatus>
   </initializationStatuses>
</results>
```



CHAPTER 32

Instance API

The Instance API is used during a Packaged CCE installation to select the name of the facility and instance for the deployment.

URL

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

Operations

- create: Creates one instance. Only one instance can be created.
- delete: Deletes one instance.
- get: Returns one instance, using the URL `https://<server ip address>/unifiedconfig/config/instance/1`.
- list: Retrieves the instance from the database.
- update: Updates one instance.

Note The user performing the operation must be in the setup group for the instance Organizational Unit that is specified.

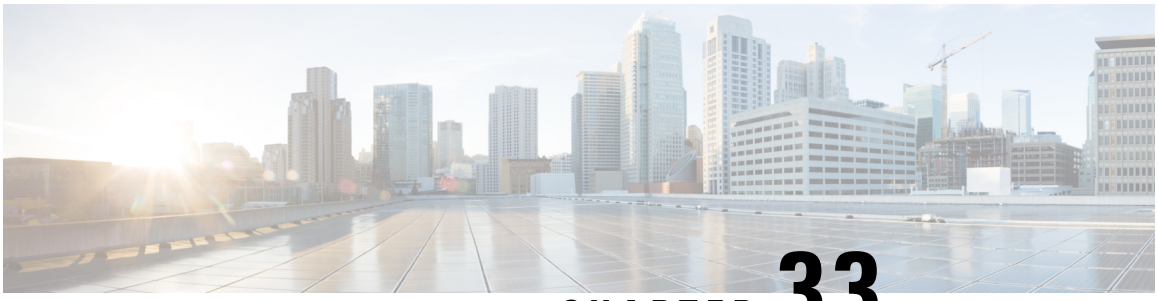
Parameters

- refURL: The refURL of the instance. See Shared Parameters, on page 8.
- facilityName: Name of an existing facility Organizational Unit in Active Directory. See Facility API, on page 109.
- instanceName: Name of an existing instance Organizational Unit in Active Directory. See Facility API, on page 109.

Example Get Response

```
<instance>
   <refURL>/unifiedconfig/config/instance/1</refURL>
   <facilityName>Lab</facilityName>
```

```
   <instanceName>pcce</instanceName>
</instance>
```



CHAPTER 33

Internet Script Editor API

The Internet Script Editor API indicates whether Internet Script Editor is enabled. If Internet Script Editor is enabled in Web Setup, the API displays the download link in the format https://<server>/install/iScriptEditor.exe.

URL

`/internetscripteditor`

Operations

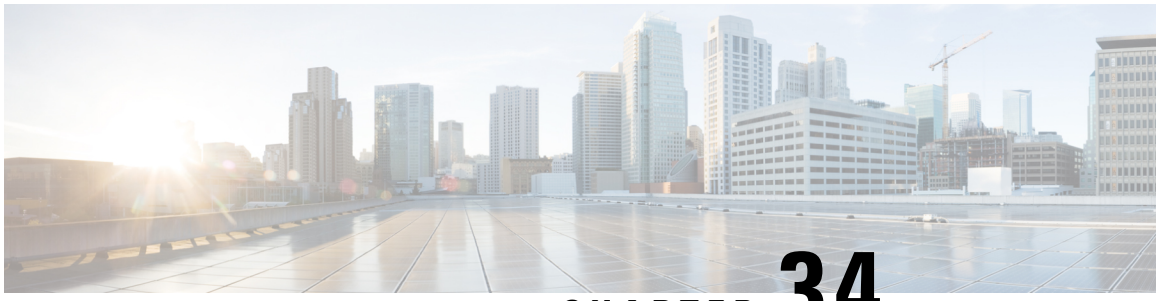
- get: Returns whether Internet Script Editor is enabled and a download link.

Parameters

- enabled: Indicates whether Internet Script Editor is enabled in Web Setup. True or false.
- downloadLink: The download link for Internet Script Editor. This link appears only when the enabled parameter is true.

Example Get Response

```
<internetScriptEditor>
    <enabled>true</enabled>
    <downloadLink>https://10.10.10.207/install/iScriptEditor.exe</downloadLink>
</internetScriptEditor>
```



CHAPTER 34

Inventory Import API

The Inventory Import API allows to import inventory during fresh install, and after deployment initialization. This is for Packaged CCE 4000 Agents and 12000 Agents deployment.

Operations

- create: Creates a new data center and peripheral set, and stores it in the database.
  - Main Data Center:
    `https://<server>:<serverport>/unifiedconfig/config/inventory/datacenter`
  - Remote Data Center:
    `https://<server>:<serverport>/unifiedconfig/config/inventory/datacenter`
  - Peripheral Set:
    `https://<server>:<serverport>/unifiedconfig/config/inventory/datacenter/<Main/datacenter_name>/peripheralset`

  Note To search for peripheral sets that belong to a specific type of PG in a Main or remote date center, use the below URL
  `https://<server>:<serverport>/unifiedconfig/config/inventory/datacenter/<Main/datacenter_name>/peripheralset?q=pgType:<UCM/VRU/MR>`

- get: Returns the list of data centers and peripheral sets using the URL:
  - Data Center: `https://<IP>/unifiedconfig/config/inventory/datacenter`. The data centers contain the peripheral sets.
  - Peripheral Set: `https://<IP>/unifiedconfig/config/inventory/peripheralset`
- delete: Deletes data center or machine or peripheral set from the database.
  - Remote Data Center:
    `https://<server>:<serverport>/unifiedconfig/config/inventory/datacenter/<datacenterid>?async=<true/false>`
  - Machine: `https://<server>:<serverport>/unifiedconfig/config/inventory/`

`datacenter/<dc_name>/machine/id` where id = machineHostID

- Peripheral Set: `https://<IP>/unifiedconfig/config/inventory/peripheralset/id` where id = peripheralSetID

CSV File Structure

For more information, see the latest PCCE Administration and Configuration Guide available at https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html.

Parameters

- name: The name of the data center.

Note Do not use system reserved terms like the core and main.

- inventoryFile: This is the CSV file that contains records of machines used in a deployment.
- content: The content of the CSV file.

Note

- To get the polling status of any task, go to `GET /unifiedconfig/inventory/results?taskId=<task_id>`
- If the name field in the payload is set to *Main* (case-sensitive), then the operation is performed on the Main data center. If not, a new remote data center is created.

Advanced Search Parameters

- **pgType:** Returns all the peripheral sets that belong to the specified pgType: UCM, VRU or MR. You can specify only one pgType at a time.

Example XML Request

Main Data Center

```
<datacenter>
   <name>Main</name>
   <inventoryFile>
       <content>
            [filecontent]
       </content>
   </inventoryFile>
</datacenter>
```

Remote Data Center

```
<datacenter>
   <name>Site1</name>
        <name>
              site1.csv
        </name>
   <inventoryFile>
       <content>
            [filecontent]
       </content>
   </inventoryFile>
</datacenter>
```

Peripheral Set

```
<datacenter>
  <inventoryFile>
     <content>
       [filecontent]
     </content>
  </inventoryFile>
</datacenter>
```

Machine

```
<inventoryFile>
  <name>
       machine.csv
  </name>
     <content>
       [filecontent]
     </content>
</inventoryFile>
```

• Templates , on page 125

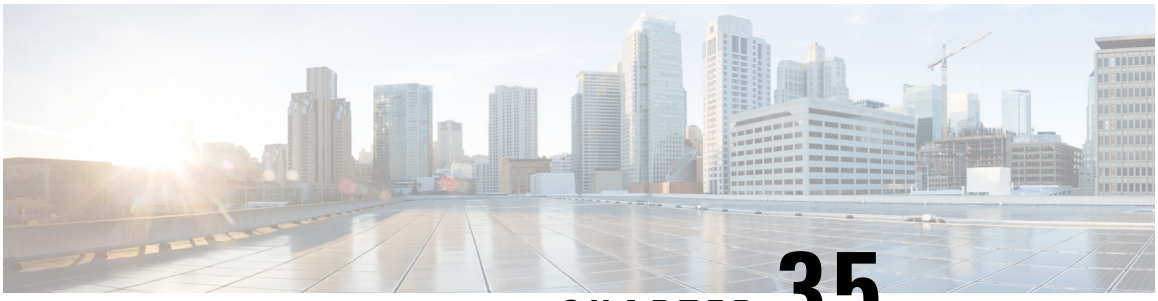
Templates

URL

https://<server>:<serverport>/unifiedconfig/config/inventory/templates/{Main/datacenter/peripheralset/machine}

HTTP Method

Use HTTP GET to submit a request to the Operation API.



CHAPTER 35

Location API

The Location feature is used to route calls locally to the agent available in the branch office, rather than routing calls to central or main office over a WAN link. In Cisco Unified Call Manager (CUCM), locations are created to implement the call admission control. The call admission control helps to regulate audio quality and video availability by limiting the amount of bandwidth that is available for audio and video calls over links between the locations.

The Location API is used to fetch or create new locations, and update the configuration by assigning the Location code and Ingress Gateway.

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

Operations

- create: Creates a new location.
- delete: Permanently deletes one location.
- get: Returns one location, using the URL `https://<server>/unifiedconfig/config/location/<id>`.
- list: Retrieves a list of locations.
- update: Updates a location.

Parameters

- refURL: Required. The refURL of the location configured in Packaged CCE. See Shared Parameters, on page 8.
- locationName: Required. The name of the location configured in CUCM. See Shared Parameters, on page 8.
- description: Optional. See Shared Parameters, on page 8.
- datacenters: The datacenters of the location. For remote sites, returns a reference to the data center, including the refURL and name.
- locationCode: Required. A unique location code that is prefixed or suffixed to the ICM label for routing the calls to the gateway.
- ucmPrimaryKey: Required. Location Primary Key ID fetched from CUCM that is used by CVP to create unique identifier for the location.

- cucmHostAddress: The IP address of the CUCM Publisher device where locations are configured.
- changeStamp: See Shared Parameters, on page 8.
- gateways: Gateway associated with the location. You can add only one gateway to a location.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• locationName • description	• locationName • cucmHostAddress

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

The Location API also supports advanced search parameters, such as gateway and datacenters.

- **gateway:<IPaddress/hostname>**: Finds the location associated to the specified hostname or IP address of the gateway. The hostname is fully matched (case-insensitive, no partial matches).
- **datacenter:<dcName>** Returns all locations which belong to the specified data center. You can specify only one data center. The data center name is fully matched (case-insensitive, no partial matches). Searching for "core" returns all locations in the core data center.

Example Get Response

```
<location>
    <refUrl>/unifiedconfig/config/location/6000</refurl>
    <locationName>Location1</locationName>
    <description>First Location</description>
    <datacenters>
        <datacenter>
            <name>Core</name>
        </datacenter>
        <datacenter>
            <name>Site1</name>
            <refUrl>/unifiedconfig/config/datacenter/5003</refUrl>
        </datacenter>
        <datacenter>
            <name>Site2</name>
            <refUrl>/unifiedconfig/config/datacenter/5004</refUrl>
        </datacenter>
    </datacenters>
    <locationCode>22</locationCode>
    <ucmPrimaryKey>30</ucmPrimaryKey>
    <cucmHostAddress>10.78.26.78</cucmHostAddress>
    <changeStamp>1</changeStamp>
    <gateways>
        <gateway>
            <address>cce-gw1.cisco.com</address>
            <refUrl>/unifiedconfig/config/datacenter/5005</refUrl>
         </gateway>
         <gateway>
            <address>cce-gw2.cisco.com</address>
```

```
            <refUrl>/unifiedconfig/config/datacenter/5006</refUrl>
          </gateway>
      </gateways>
  </location>
```

Following are the REST responses received during execution of REST API to configure the location:

- **Success** - Configuration changes persist in AW DB and synchronized with respective devices.

  Code: 200

  Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with one or more devices.

  Code: 201

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with CVP failed.)

  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

```
<apiErrors>
<apiError>
<errorMessage>Configuration update failed for one or more devices.</errorMessage>
<errorType>PARTIAL_SUCCESS</errorType>
</apiError>
</apiErrors>
```

- **Failure**- The configuration updates to AW DB is failed.

- Location Properties API, on page 129

Location Properties API

The Location Properties feature provides options for the placement of the location routing code. The location routing code is configured using the Location API . See Location API, on page 127.

You can place the routing code at the beginning of the Network VRU label, in the middle of the Network VRU label and the correlation ID, or can choose not to insert the routing code. The Location Properties are the global settings.

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

Operations

- get: Returns the location property data, using the URL `https://<server>/unifiedconfig/config/locationproperties`.
- update: Updates a location property data.

Parameters

- locationRoutingCodeInsertOption: Required. Placement of the location routing code. As this is a global setting, any change in the value applies to all CVP across all datacenter. The following are the key values:
  - beginning - Location routing code at the beginning of the Network VRU label.
  - middle - Location routing code between the Network VRU label and the correlation ID. This is the default value.
  - none - Do not insert the location routing code.

  Default value is middle.

Example Get Response

```
<CVP>
<locationProperties>
<locationRoutingCodeInsertOption>middle</locationRoutingCodeInsertOption>
</locationProperties>
</CVP>
```

Following are the REST responses received during execution of REST API to configure the location properties data across all sites:

- **Success** - Configuration changes persist in AW DB and synchronized with respective devices.

  Code: 200

  Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with one or more devices.

  Code: 201

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with CVP Call Server failed.)

  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

  ```
  <apiErrors>
  <apiError>
  <errorMessage>Configuration update failed for one or more devices.</errorMessage>
  <errorType>PARTIAL_SUCCESS</errorType>
  </apiError>
  </apiErrors>
  ```

- **Failure**- The configuration updates to AW DB is failed.

CHAPTER 36

Log Collection API

Use the Log Collection API to collect the log files, or request a download of the log files in one collected zipped file.

Operations

- create: Creates one request to collect the log files, and begins collecting them.
  - Only one log collection may be performed at a time.
  - The maximum number of saved log collections is 3.
  - Inventory status errors must be cleared before log collection starts.
- delete: Deletes one log collection.
- get: Returns the log collection item, using the URL `https://<server>/unifiedconfig/config/logcollection/<id>`.
- list: Retrieves a list of collection requests.

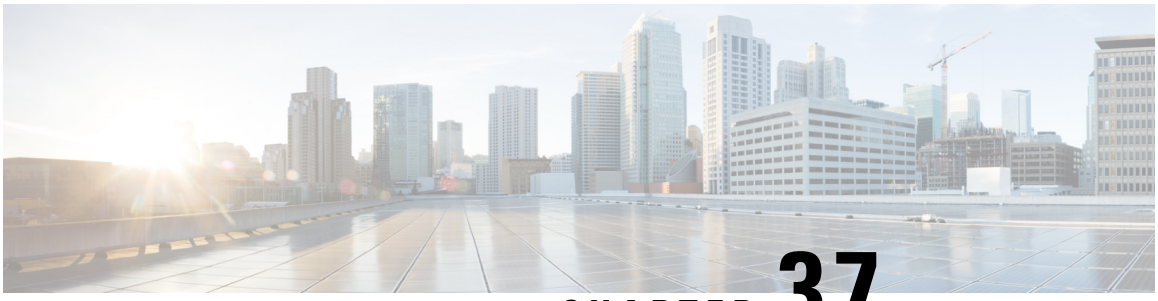
Parameters

- refURL: The refURL of the log collection. See Shared Parameters, on page 8.
- startDateTime: The start date of the logs collected.
- endDateTime: The end date of the logs collected.
- description: See Shared Parameters, on page 8.
- components: A list of components for which logs are collected. Defaults to all components if the list is blank or not provided. Possible component values include:
  - 1: Unified Contact Center Enterprise (CCE)
  - 2: Unified Customer Voice Portal (CVP)
  - 3: Unified Communications Manager (CM)
  - 4: Cisco Unified Intelligence Center (Intelligence Center)
  - 5: Finesse

- status.state: The status of the collection request: IN_PROGRESS, DONE, or ERROR.
- status.apiErrors: The error indicating why the collection request failed. Returned when status.state is ERROR.
- resultsFile: Zipped file containing logs collected. Includes the following parameters:
  - refURL: The URL of the zipped file which is used for download.
  - size: The size of the zipped file (in bytes).

Example Get Response

```
<logCollection>
    <refURL>/unifiedconfig/config/logcollection/1</refURL>
     <status>
         <state>IN_PROGRESS</state>
     </status>
     <components>
          <component>1</component>
          <component>2</component>
     </components>
     <description>this is a log collection to see if ____</description>
     <startDateTime>1368564152000</startDateTime>
     <endDateTime>1368564156000</endDateTime>
     <resultsFile>
        <refUrl>/unifiedconfig/config/logcollection/(id)/log</refUrl>
        <size>450</size>
     </resultsFile>
</logCollection>
```



CHAPTER 37

Machine Inventory API

This API returns the machines in the solution. Machines include VMs, VM Hosts, external machines, and gateways.

For information on how to perform a machine inventory scan, see Scan API, on page 197.

URL

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

Operations

- create: Creates a machine by updating the database. See table below for restrictions per machine type. Create is allowed for external machines types only.
- delete: Removes one machine.

Note

- You cannot delete the Virtualized Voice Browser (VVB), Unified SIP Proxy and Gateway external machines if they are associated with a SIP Server Group. To delete these external machines, you must disassociate them from the SIP Server Group.

- get: Returns one machine and all associated addresses and services based on machine ID, using the URL `https://<server>/unifiedconfig/config/machineinventory/<id>`.
  - status: Returns any alerts indicating errors in the state of the inventory, using the URL `https://<server>/unifiedconfig/config/machineinventory/status`.
  - stats: Returns CVP Call Server and VXML Server statistics based on the machine type, using the URL `https://<server>/unifiedconfig/config/machineinventory/<machine_id>/stats`.
- list: Retrieves a list of all machines in the inventory. See table below for more details.
- update: Updates one machine.

Type	Create / Update / Delete operations allowed	Number allowed
VM_HOST	No	1 Side A 1 Side B
CCE_ROGGER (Applicable only for 2000 Agents and 4000 Agents)	No	1 Side A 1 Side B
CCE_ROUTER (Applicable only for 12000 Agents)	No	1 Side A 1 Side B
CCE_LOGGER (Applicable only for 12000 Agents)	No	1 Side A 1 Side B
CCE_PG	No	1 Side A 1 Side B
CCE_AW	Side A: Update only Side B: No	1 Side A 1 Side B
CVP	No	1 Side A 1 Side B
CM	Update only	0 - Must be changed after initial scan.
CM_PUBLISHER	Update only	1 Side A, 0 Side B for on box CM Deployments 0 for off box CM deployments
CM_SUBSCRIBER	Update only	1 Side A, 1 Side B for on box CM Deployments 0 for off box CM deployments
CVP_REPORTING	No	0 - 1 Side A
CUIC_PUBLISHER	Update only	1 Side A
CUIC_SUBSCRIBER	No	1 Side B
CVP_OPS	Update only	1 Side A
FINESSE	Side A: Update only Side B: No	1 Side A 1 Side B 0 - 1 Side B
ECE	No	

Type	Create / Update / Delete operations allowed	Number allowed
EXTERNAL_CVVB (Cisco Virtualized Voice Browser)	All	Supports 0 to unlimited number of external CVVB machines for the main site.
EXTERNAL_GATEWAY	All	Supports 0 to unlimited number of external Gateways for the main site.
EXTERNAL_CUSP Cisco Unified SIP Proxy (CUSP)	All	Supports 0 to unlimited number of external CUSP Servers for the main site.
EXTERNAL_SOCIAL_MINER)	All	0 - 1
EXTERNAL_CM_PUBLISHER	All	0 - 10 for on box CM deployments 1 - 11 for off box CM deployments
EXTERNAL_CM_SUBSCRIBER	No	External subscribers cannot be created, updated, or deleted as this automatically occurs when the external publisher is created, updated, or deleted.
EXTERNAL_CVP_REPORTING	All	0 - 1
EXTERNAL_HDS	Update only	0 - 2
EXTERNAL_MEDIA_SENSE	All	0 - 1
EXTERNAL_ECE	All	0 - 1
EXTERNAL_THIRD_PARTY _MULTICHANNEL	All	0 - 1
EXTERNAL_MEDIA_SERVER	All	No limit
DATA_CENTER	No create, update, or delete	0 - 10
DC_CCE_PG	No create, update, or delete	0 - 20
DC_CVP	No create, update, or delete	0 - 20
DC_FINESSE_PRIMARY	Update only	0 - 10
DC_FINESSE_SECONDARY	No create, update, or delete	0 - 10
DC_EXTERNAL_CUSP	All	Supports 0 to unlimited number of CUSP Server for the remote site.

Type	Create / Update / Delete operations allowed	Number allowed
DC_EXTERNAL_GATEWAY	All	Supports 0 to unlimited number of Gateways for the remote site.
DC_EXTERNAL_CVVB	All	Supports 0 to unlimited number of CVVB machines for the remote site.
DC_EXTERNAL_CVP_REPORTING	All	0 - 1
DC_EXTERNAL_MEDIA_SERVER	All	No limit
ECE_WEB_SERVER	Yes	No limit
DC_ECE_WEB_SERVER	Yes	No limit

Parameters

Machine parameters:

- refURL: The refURL of the machine. See Shared Parameters, on page 8.
- name: External name of the machine. For example, the VM host name. Valid characters are period (.), hyphen (-), underscore (_), and alphanumeric. The first character must be alphanumeric. Maximum length is 128 characters.
- changeStamp: See Shared Parameters, on page 8.
- type: The type of machine.
- versionInfo: Version info available depending on product type. Includes the following parameters (by product type):
  - UCCE: version, buildNumber, esNumber, patchVersion
  - CVVB: version and buildNumber
  - CVP: version, buildNumber, buildDate, esNumber, srNumber, dropNumber
  - VOS: version and buildNumber
  - Gateway: version
- vmhost: A reference to a machine of type VM_HOST, including refURL and name. See References, on page 5.
- autogenerated: Indicates if the information was generated automatically.
- hostName: The host name of the machine.
  - If you provide a hostname, a lookup is performed to find the FQDN for that host. If the FQDN is found, the hostName field is then set to that FQDN value and the Machine Address is set to the IP address corresponding to the hostname.
  - If you do not provide a hostname, a lookup is performed using the address field in the API request. If the FQDN is found, the hostName field is set to that FQDN value.

- networks: A collection of network parameters. See the network parameters below.
- vmInstanceUuid: A unique identifier for the virtual machine.

Network parameters:

- type: Public or private. Private networks are only specified for CCE_PG and CCE_ROGGER.
- address: The IP address. Must be valid hostname, IPV4, or IPV6 address.
- services: A collection of service parameters. See the services parameters below.

Services parameters:

- uri: The service URL, for example, Diagnostic Portal URL or management console link.
- port: The port for this service.
- password: The password used to access the service. The password can be used when creating or updating, but is not returned.
- enablePassword: Used for gateways.
- description: The description of the service. See Shared Parameters, on page 8.
- pairing: Indicates if services on different machines are related. Related services have a matching value.

  For the service type **PRINCIPAL_AW**, the pairing value is "true" for the AW that manages credentials for the features like Context Service, and "false" for all other AWs.

  For the service type **FTP_CREDENTIAL**, the following pairing values are supported:

  - True: Indicates FTP is enabled on the Media Server.
  - False: Indicates FTP is disabled on the Media Server.

- username: The username used to access the service. Username maximum is 128 characters.

  For the service type **FTP_CREDENTIAL**, the userName *Anonymous* indicates that anonymous access is enabled on the FTP Server.

- status: Used to indicate the status of the machine which are of type CVP_WSM, DIAGOSTIC_PORTAL, or CVVB.
- type: The service type. Values for type are as follows:
  - ESXI: VM Host running ESXi server for connection to interrogate for VMs.
  - AXL: AXL connection information for Unified CM. Required for UCM Publisher.
  - DIAGNOSTIC_PORTAL: For CCE and Unified CVP, this is the Diagnostic Portal API. For Finesse and Unified Intelligence Center, this is the SOAP API. Required for CCE_AW on side A, CVP Reporting Server, CUSP and FINESSE on side A, and CUIC Publisher.
  - CVP_WSM: Connection information for CVP Servers.
  - FTP_CREDENTIAL: Connection information for FTP service on the Media Server.
  - GATEWAY: Connection information for gateways.
  - CVVB: Provides connection information of the external CVVB machines.

- MANAGEMENT_LINK: Provides a URL of the management console for the machine.
- TIP_PG: LiveData connection information for peripheral gateways.
- TIP_ROUTER: LiveData connection information for the router.
- ADMINISTRATION: Connection information for administrative access. Required for CVP OAMP.
- SM_REST_API: SocialMiner REST API.
- CONTEXT_SERVICE: Contains the last-written Context Service connection data.
- IDS: Indicates an Identity Server service. IDS contains the following parameters:
  - components: Information about the component, including the component refURL and component name.
- IDS_PRIMARY_REF: When present, the <uri> parameter for this service refers to the primary Identity Server (IdS) to which this component is associated.
- IDS_SECONDARY_REF: Read only. The <uri> parameter for this service refers to the backup IdS to which this component is associated. For a create or update request, this field is automatically filled with the refURL of the secondary IdS (if present).
- MR_PG_CONNECTION: Connection information for the MR PG.
- VRU_PG_CONNECTION: Connection information for the VRU PG.
- UCM_PG_CONNECTION: Connection information for the UCM PG.
- DATA_CENTER_REF: Read only. Contains information for the Data Center machine.
- PRINCIPAL_AW: Indicates the AW that manages credentials for the features like Context Service.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description • hostName	• name (default) • description • hostName

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

You can perform a machine type search or a service type search on the Machine Inventory API:

- **types: (machine_type1 | machine_type2 | machine_type3...)**: Returns all the machines of the specified type. For example, **types:(VM_HOST | EXTERNAL_SOCIAL_MINER | PUB_SUBSCRIBER)** returns all the machines that belong to any of the specified machine types. The machine type is case-insensitive.

- **datacenters: (dc1|dc2|dc3...)**: Returns machines which belong to any of the specified data center. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all machines in the core data center.
- **serviceType:<service type>**: Returns only machines associated with the specified service type. For example, **serviceType:IDS_PRIMARY_REF** returns only SSO-capable machines.
- **peripheralsets: (ps1|ps2|ps3...)**: Returns machines which belong to any of the specified peripheral sets. The peripheral set names are fully matched (case-insensitive, no partial matches).
- **ipaddress: <IP address>**: Returns machines that are configured with the specified IP address in the inventory. You can even specify a partial IP address to search. Search includes both public and private IP addresses.

Example Inventory Status Response

```
https://<server>/unifiedconfig/config/machineinventory/status
```

```
<status>
   <alerts>
      <alert>
         <apiErrors>
            <apiError>
               <errorData>CM_PUBLISHER</errorData>
               <errorMessage>CM_PUBLISHER not found on vmhost sideA</errorMessage>
               <errorType>inventory.MissingMachine</errorType>
            </apiError>
         </apiErrors>
         <machine>
            <host>sideA</host>
            <type>CM_PUBLISHER</type>
         </machine>
      </alert>
   </alerts>
   <scanInfo>
      <lastScanDateTime>1374842924017</lastScanDateTime>
      <scanState>Idle</scanState>
   </scanInfo>
</status>
```

Example Stats Response

```
https://<server>/unifiedconfig/config/machineinventory/<machine_id>/stats
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<statistics>
    <callServer>
        <icmStats>
         ....
        </icmStats>
        <infrastrutureStats>
         ....
        </infrastrutureStats>
        <sipStats>
         ...
        </sipStats>
    </callServer>
    <vxmlServer>
        <infraStructure>
         ...
        </infraStructure>
```

```
        <ivr>
         ...
        </ivr>
        <vxml>
         ...
        </vxml>
    </vxmlServer>
</statistics>
```

Example Get Response

```
https://<server>/unifiedconfig/config/machineinventory/<id>
```

```
<machine>
    <changeStamp>4</changeStamp>
    <refURL>/unifiedconfig/config/machineinventory/12073</refURL>
    <networks>
        <network>
            <address>10.10.10.21</address>
            <services>
                <service>
                    <autoGenerated>false</autoGenerated>
                    <port>7890</port>
                 <status>OUT_OF_SYNC</status>
                    <type>DIAGNOSTIC_PORTAL</type>
                    <userName>user@domain</userName>
                </service>
            </services>
            <type>PUBLIC</type>
        </network>
    </networks>
    <autoGenerated>false</autoGenerated>
    <hostName>CCE-AW-1-21</hostName>
    <type>CCE_AW</type>
    <versionInfo>
        <buildNumber>03297</buildNumber>
        <version>11.0(1)</version>
    </versionInfo>
    <name>WJ2-AW-1</name>
    <vmHost>
        <refURL>/unifiedconfig/config/machineinventory/12071</refURL>
        <name>sideA</name>
    </vmHost>
    <vmInstanceUuid>50290e07-fa3d-2667-b84b-1c35989000e2</vmInstanceUuid>
</machine>
```

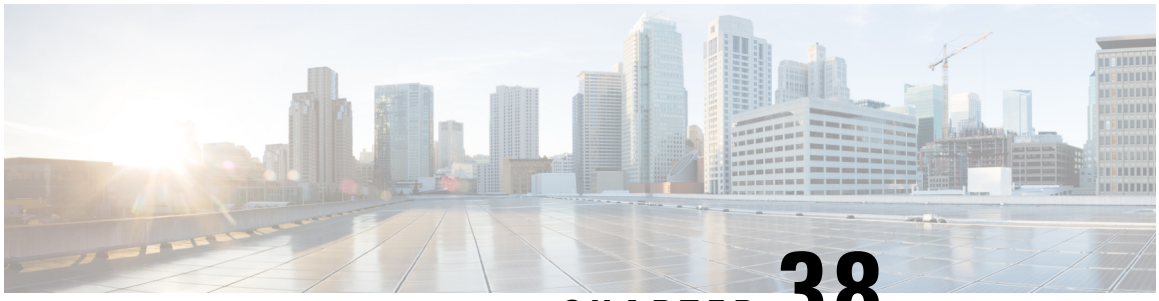
Example Updating SSO-Capable Components Associated to an IdS

SSO-capable components can be associated to an IdS with the <components> element. For example, a PUT request like this will update the IDS_PRIMARY_REF and IDS_SECONDARY_REF of any SSO-capable components. Note that refURLs of any on-box PCCE machines will be ignored as these cannot be changed. An example of an off-box PCCE machine is the EXTERNAL_HDS machine.

This operation will also de-associate any existing component references to that IdS that aren't in the list. For example, if there were three components associated to this IdS before the operation, only two components would be associated after the operation was complete.

```
<machine>
    <changeStamp>1</changeStamp>
    <refURL>/unifiedconfig/config/machineinventory/5000</refURL>
    <networks>
        <network>
            <type>PUBLIC</type>
```

```
            <address>10.10.10.22</address>
            <services>
                <service>
                    <type>IDS</type>
                    <components>
                        <component>
                            <refURL>/unifiedconfig/config/machineinventory/6000</refURL>
                        </component>
                        <component>
                            <refURL>/unifiedconfig/config/machineinventory/6001</refURL>
                        </component>
                    </components>
                </service>
            </services>
        </network>
    </networks>
</machine>
```



CHAPTER 38

Media Routing Domain API

A media routing domain is a collection of skill groups associated with a common media class. It is used to organize how requests for different media are routed.

Use the Media Routing Domain (MRD) API to list the MRDs currently defined in the database, define new MRDs, and view, edit, and delete existing MRDs.

The built-in Cisco_Voice MRD and legacyMultichannel MRDs are read-only; they cannot be created, updated, or deleted. You can perform all API operations on multichannel MRDs.

URL

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

Operations

- create: Creates an MRD.
- delete: Permanently deletes one MRD.
- list: Retrieves a list of MRDs.
- get: Returns one MRD using the URL `https://<server>/unifiedconfig/config/mediaroutingdomain/<id>`.
- update: Updates one MRD.

Parameters

- refURL: The refURL of the MRD. See Shared Parameters, on page 8.
- name: Name of the MRD. See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- id: The database id of the MRD. Read-only field.
- type: The type of MRD. Values are as follows:
  - voice: Used only for the built-in Cisco_Voice MRD. These MRDs are read-only.
  - legacyMultichannel: Used for MRDS for the Enterprise Chat and Email application. These MRDs are read-only.

    - multichannel: (Default) Used for MRDs for Task Routing APIs.

- taskLife: If the connection goes down, the amount of time, in seconds, that the system waits before ending all tasks. Default is 1200.
- taskStartTimeout: The amount of time, in seconds, that the system waits between an agent being selected for a task and an agent being offered or beginning the task. When this time is reached, the system makes the agent Not Routable. Default is 30.
- maxTaskDuration: The maximum duration for a task, in seconds. Default is 28800.
- serviceLevelThreshold: Maximum time in seconds that a customer should wait before being connected with an agent. Default is 30.
- interruptible: Indicates if an agent can be interrupted by assigned tasks from another MRD. Values are true/false.
- maxTasksInQueue: The maximum number of tasks allowed to be queued at one time.
- maxTimeInQueue: The maximum amount of time, in seconds, a task can be queued.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description • id	• name (default) • description • id • interruptible • maxTasksInQueue • maxTaskDuration • maxTimeInQueue • serviceLevelThreshold • taskLife • taskStartTimeout

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

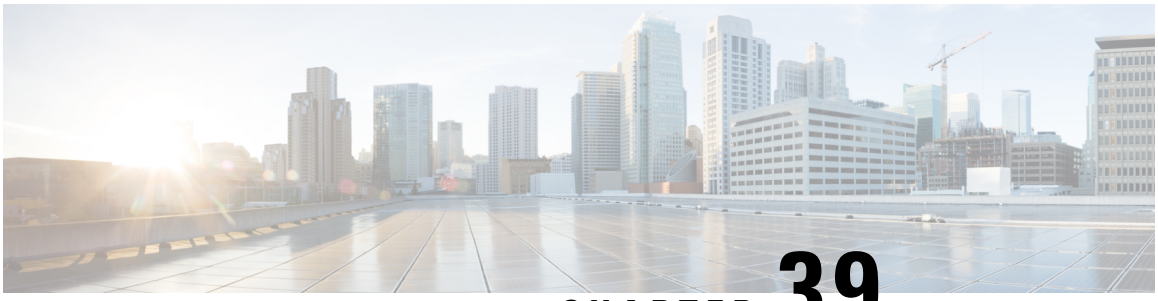
You can perform a nonVoiceOnly search on the Media Routing Domain API:

- **nonVoiceOnly**: Set this attribute to true in the search query parameter to make the API return only media routing domains other than the Cisco_Voice MRD. For example, **q=nonVoiceOnly:true**.

Example Get Response

```
<mediaRoutingDomains>
  <mediaRoutingDomain>
     <changeStamp>0</changeStamp>
     <refURL>/unifiedconfig/config/mediaroutingdomain/5001</refURL>
     <description>Media channel for routing Chat tasks</description>
     <id>5001</id>
     <type>multichannel</type>
```

```
    <interruptible>false</interruptible>
    <taskLife>1200</taskLife>
    <taskStartTimeout>30</taskStartTimeout>
    <maxTaskDuration>28800</maxTaskDuration>
    <maxTasksInQueue>1000</maxTasksInQueue>
    <maxTimeInQueue>1000</maxTimeInQueue>
    <name>Chat_Task_MRD</name>
    <serviceLevelThreshold>30</serviceLevelThreshold>
 </mediaRoutingDomain>
 <mediaRoutingDomain>
    <changeStamp>0</changeStamp>
    <refURL>/unifiedconfig/config/mediaroutingdomain/1</refURL>
    <description>Default Media Routing Domain for Cisco_Voice</description>
    <id>1</id>
    <type>voice</type>
    <interruptible>false</interruptible>
    <taskLife>1200</taskLife>
    <taskStartTimeout>30</taskStartTimeout>
    <maxTaskDuration>28800</maxTaskDuration>
    <name>Cisco_Voice</name>
    <serviceLevelThreshold>30</serviceLevelThreshold>
  </mediaRoutingDomain>
</mediaRoutingDomains>
```



CHAPTER 39

Network VRU Script API

Calls may be sent to a Voice Response Unit (VRU) instead of or before they are sent to an agent. In the Packaged CCE deployment, the VRU is Customer Voice Portal (Unified CVP). You must configure network VRU scripts to direct Unified CVP on how to handle the treatment of individual calls.

Use the Network VRU Script API to list, create, edit and delete network VRU scripts.

URL

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

Operations

- create: Creates one network VRU script.
- delete: Deletes one network VRU script from the database.
- get: Returns one network VRU script, using the URL `https://<server>/unifiedconfig/config/networkvruscript/<id>`.
- list: Retrieves a list of network VRU scripts.
- update: Updates one network VRU script.

Parameters

- refURL: The refURL of the network VRU script. See Shared Parameters, on page 8.
- name: The name of the network VRU as seen by CCE. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- routingType: This field is optional and defaults to 1. Options are:
  - 1: Voice. Used by Unified CVP.
  - 2: Multichannel. Used by Email and Web Collaboration.
- vruScriptName: Required. The name of the script as it is known on the Unified CVP. Maximum length of 39 characters allowed.

- timeout: Number of seconds for the system to wait for a response from the routing client after directing it to run the script. Must be an integer that is 1 or higher. Default is 180.
- configParam: Optional string used by Unified CVP to pass additional parameters to the IVR Service. Maximum length is 255 characters.
- interruptible: Indicates whether the script can be interrupted. Values are true/false.

Search and Sort Values

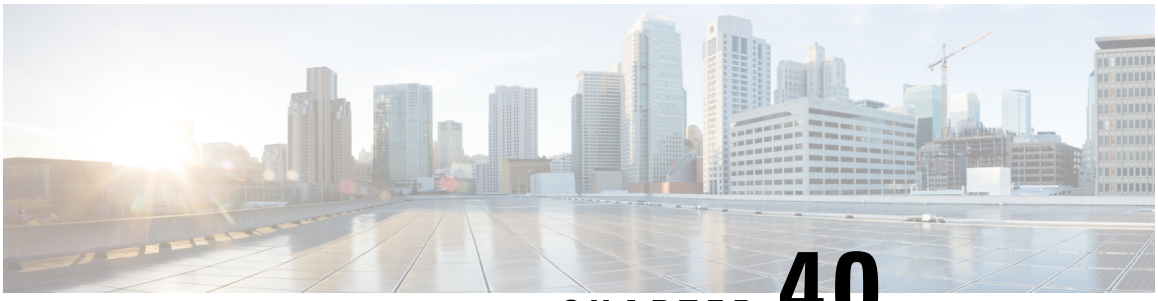
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description • vruScriptName • timeout • configParam • interruptible

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<networkVruScript>
    <refURL>/unifiedconfig/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>
```



CHAPTER 40

Operation API

Use the Operation API to save changes to several items of the same type in a single request. The following changes are allowed in an operation:

- delete: Multiple items of the same type. Any item that supports the delete operation can be deleted using the Operation API.
- Agent update: Update multiple agents.

URL

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

HTTP Method

Use HTTP POST to submit a request to the Operation API.

Parameters

- operationType: Indicates if the items specified in the refURLs should be updated or deleted. Values are update/delete.
- refURLs: A collection of refURL parameters indicating which items are included in the request. See Shared Parameters, on page 8.
- changeset: Includes the parameters that are changed in an update operation. See Agent API, on page 17.
  - skillGroupsAdded
  - skillGroupsRemoved
  - description
  - agentTeam
  - agentDeskSettings

Example Delete Request

```
<operation>
  <operationType>delete</operationType>
  <refURLs>
```

```
    <refURL>/unifiedconfig/config/calltype/5000</refURL>
    <refURL>/unifiedconfig/config/calltype/5001</refURL>
  </refURLs>
</operation>
```

Example Update Request

```
<operation>
  <operationType>update</operationType>
  <refURLs>
    <refURL>/unifiedconfig/config/agent/5000</refURL>
    <refURL>/unifiedconfig/config/agent/5001</refURL>
  </refURLs>
  <changeSet>
    <agent>
      <description>New description</description>
      <agentDeskSettings>
        <refURL>/unifiedconfig/config/agentdesksetting/5434</refURL>
      </agentDeskSettings>
      <agentTeam>
        <refURL>/unifiedconfig/config/agentteam/5000</refURL>
      </agentTeam>
      <skillGroupsAdded>
        <skillGroup>
          <refURL>/unifiedconfig/config/skillgroup/6000</refURL>
        </skillGroup>
      </skillGroupsAdded>
      <skillGroupsRemoved>
        <skillGroup>
          <refURL>/unifiedconfig/config/skillgroup/6001</refURL>
        </skillGroup>
      </skillGroupsAdded>
      </agent>
  </changeSet>
</operation>
```

Response Parameters

- status: Indicates the state of the operation.
  - success: The operation succeeded for all items.
  - partialSuccess: The operation succeeded for some items, but other items had errors.
  - failure: The operation failed for all items.
- apiErrors: Errors indicate which items had errors and the cause of the error.

Example Success Response

The following example shows the response when the delete operation is successful:

```
<operationsResult>
  <status>success</status>
</operationsResult>
```

Example Partial Success Message

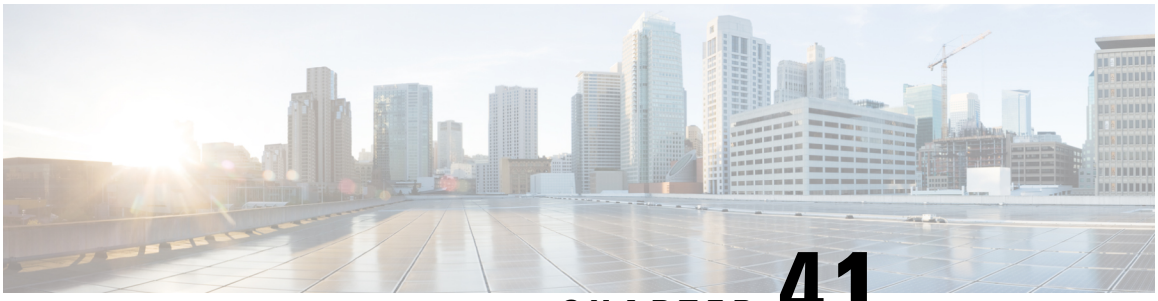
The following example shows a partial success response for a request to delete several agents:

```
<operationsResult>
    <apiErrors>
        <apiError>
            <errorDetail xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type=
             "resourceErrorDetail">
                <refURL>agent/1</refURL>
                <apiErrors>
                    <apiError>
                        <errorMessage>The specified ID does not exist
                         in the database.</errorMessage>
                        <errorType>notFound.dbData</errorType>
                    </apiError>
                </apiErrors>
            </errorDetail>
            <errorMessage>There were one or more errors processing the following
             request: delete agent/1</errorMessage>
            <errorType>operation.resourceErrors</errorType>
        </apiError>
        <apiError>
            <errorDetail xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type=
             "resourceErrorDetail">
                <refURL>agent/2</refURL>
                <apiErrors>
                    <apiError>
                        <errorMessage>The specified ID does not exist
                         in the database.</errorMessage>
                        <errorType>notFound.dbData</errorType>
                    </apiError>
                </apiErrors>
            </errorDetail>
            <errorMessage>There were one or more errors processing the following
             request: delete agent/2</errorMessage>
            <errorType>operation.resourceErrors</errorType>
        </apiError>
    </apiErrors>
    <status>partialSuccess</status>
</operationsResult>
```

Example Failure Response

The following example shows a failure response for a request to delete a call type that does not exist:

```
<operationsResult>
  <status>failure</status>
  <apiErrors>
    <apiError>
      <errorDetail xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type=
       "resourceErrorDetail">
        <refURL>/unifiedconfig/config/calltype/9999</refURL>
        <apiErrors>
          <apiError>
            <errorMessage>The specified ID does not exist in the database.</errorMessage>
            <errorType>notFound.dbData</errorType>
          </apiError>
        </apiErrors>
      </errorDetail>
      <errorMessage>There were one or more errors processing the following request:
       delete /unifiedconfig/config
      /calltype/9999</errorMessage>
      <errorType>operation.resourceErrors</errorType>
    </apiError>
  </apiErrors>
</operationsResult>
```



CHAPTER 41

Outbound API

Outbound API allows you to use REST APIs to create, modify, and delete Outbound Option campaigns.

Outbound API provides a streamlined mechanism for creating campaigns with a single associated query rule and import rule. As such, if you use the API to create a campaign, that campaign is not available in the Outbound Campaign tool. If a campaign was created with the API, you must use the API to view, edit, or delete it. If a campaign was created with the Outbound Campaign tool, you must use the Outbound Campaign tool to view, edit, or delete it.

Administrative scripts are not required for Outbound Option campaigns created with the Outbound API. If an administrative script is provided, the information in the script overrides the information defined in the API.

Outbound API consists of the following APIs:

- Outbound Campaign API: Use this API to define new Outbound Option campaigns, and to view, edit, or delete existing campaigns.
- Campaign Status API: Use this API to get the real-time status of running Outbound Option campaigns.
- Do Not Call API: Use this API to set the Do Not Call (DNC) import rule configuration for Outbound Option. This prevents the dialer from dialing numbers on the DNC list.
- Import API: Use this API to import customer contact information for an Outbound Option campaign.
- Personal Callback API: Use this API to configure your Outbound Option campaign to handle personal callbacks. You can create personal callback records individually or in bulk. You can also update or delete personal callback records.
- Time Zone API: Use this API to list all available time zones and to get information about a specified time zone.

- Outbound Campaign API, on page 154
- Campaign Status API, on page 161
- Do Not Call API, on page 163
- Import API, on page 165
- Personal Callback API, on page 170
- Time Zone API, on page 175

Outbound Campaign API

An Outbound Campaign makes outgoing calls to customers for a specific purpose or task and delivers these calls to agents.

Use the Outbound Campaign API to define new outbound campaigns, and to view, edit, or delete existing outbound campaigns.

Note In the Role API, when you enable the CampaignStatus or CampaignContact subfeature in the accessList parameter for a custom role then the Outbound Campaign API is provided with Update Only Access instead of Full Access. With Update Only Access, you cannot create and delete a campaign using Outbound Campaign API.

URL

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

Operations

- create: Creates one campaign and stores it in the database.
- delete: Deletes one campaign from the database. The campaign is saved under **Configuration Manager** > **Miscellaneous Tools** > **Deleted Objects**.

  You can only delete campaigns that were created using the API. You cannot use the API to delete campaigns that were created in the Outbound Option Campaign tool.
- list: Retrieves a list of campaigns from the database. Only campaigns that were created using the Outbound Option API are retrieved. Campaigns created using the Outbound Option Campaign tool do not appear in the list.
- get: Returns one campaign from the database using the URL `https://<server>/unifiedconfig/config/campaign/<id>`.

  You must specify the ID for a campaign that was created using the API. If you specify the ID for a campaign created using the Outbound Option Campaign tool, the request returns CceDBDataNotFoundException.
- update: Updates one campaign in the database using the URL `https://<server>/unifiedconfig/config/campaign/<id>`.

  You can only update campaigns that were created using the API. You cannot use the API to update campaigns that were created in the Outbound Option Campaign tool.

Parameters

- name: Required. The name of the campaign. See Shared Parameters, on page 8.

Note You cannot use the system reserved terms such as dnc and none as a campaign name.

- description: Optional description for the campaign. See Shared Parameters, on page 8.
- linesPerAgent: The number of lines dedicated to each agent in the campaign. Range is 1 to 100. Default is 1.5. This parameter performs as follows in the Outbound Option dialing modes:
  - Preview mode: Ignored (always 1).
  - Progressive mode: Used as defined.
  - Predictive mode: Used as an initial value.
- maximumLinesPerAgent: The upper bound for the number of customers the dialer dials for a reserved agent when a campaign is running in predictive mode. Range is 1 to 100. Default is 2.
- abandonEnabled: True or false. Default is true.
- abandonPercent: When enabled (abandonEnabled is set to true), you can set the abandoned calls limit for the percentage of abandoned calls in the campaign. When disabled (abandonEnabled is set to false), the campaign dials without regard to the abandon limit. Range is 1 to 100. The granularity is to one 10th of a percent. Default is 3.
- predictiveCorrectionPace: A count of the number of live voice connections that must occur before the Dialer adjusts. Increasing this number results in less frequent adjustments based on larger sample size. Decreasing this number results in more frequent adjustments using a smaller sample size. Range is 10 to 5000. Default is 70.
- predictiveGain: The size of the adjustment to lines per agent each time an adjustment is made. Increasing this number results in larger lines per agent adjustments. Decreasing this number results in smaller lines per agent adjustments. Range is 0.1 to 3.0. Default is 1.
- noAnswerRingLimit: The number of times the software allows a dialed phone number to ring. Range is from 2 to 10. Default is 4.
- maxAttempts: The maximum number of attempts, including callbacks and retries. Range is from 1 to 100. Default is 3.
- minimumCallDuration: Minimum duration (in seconds) of an outbound call. If the outbound call is less than the specified value, Outbound Option considers the call to be customer abandoned and schedules the record for a retry. To disable this feature, set the parameter to 0. Range is 0 to 10. Default is 1.
- noAnswerDelay: The time (in minutes) that the software waits before calling back a no-answer call. Range is 1 to 99999. Default is 60.
- busySignalDelay: The time (in minutes) that the software waits before calling back a busy phone number. Range is 1 to 99999. Default is 60.
- customerAbandonedDelay: If a customer abandons a call, the time (in minutes) that the Dialer waits before calling the customer back. Range is 1 to 99999. Default is 30.
- dialerAbandonedDelay: If the Dialer abandons a call, the time (in minutes) that the Dialer waits before calling the customer back. Range is 1 to 99999. Default is 60.

- answeringMachineDelay: If an answering machine answers a call, the time (in minutes) that the Dialer waits before calling the customer back. Range is 1 to 99999. Default is 60.
- customerNotHomeDelay: If a customer is not home, the time (in minutes) that the Dialer waits before calling the customer back. Range is 1 to 99999. Default is 60.
- personalizedCallbackEnabled: If enabled, this parameter allows an agent to schedule a callback to a customer for a specific date and time. A personal callback connects the same agent who initiated the callback to the customer. True or false. Default is false.
- rescheduleCallbackMode: Determines how Outbound Option handles a personal callback if the agent is not available. Default is useCampaignDN. Options are as follows:
  - Use campaign DN
  - Same time next business day
  - Abandon
- campaignPurposeType: The type of campaign. Default is agentCampaign. The options are as follows:
  - Agent Campaign: This type of campaign uses an outbound mode that causes the Dialer to transfer every customer call associated with a specific skill group to an agent.
  - IVR Campaign: This type of campaign uses an outbound mode that causes the Dialer to transfer every customer call associated with a specific skill group to a service control-based IVR instead of an agent. This option allows a contact center to run unassisted outbound campaigns using prerecorded messages in the IVR.
- ipAmdEnabled: When enabled, directs the Dialer to perform a specific action if it detects an answering machine. True or false. Default is true.
- amdTreatmentMode: If enabled (ipAmdEnabled is set to true), when the Dialer detects an answering machine, it does one of the following:
  - Abandon call (default)
  - Transfer to agent
  - Transfer to IVR route point
- ipTerminatingBeepDetect: When this parameter is set to true, the Dialer transfers the call after detecting the answering machine beep. True or false. Default is false.
- timeZone: Required. The refURL and display name for the selected time zone. The default time zone is UTC (Universal Coordinated Time). The display name is the text that may be displayed in a user interface and can be localized.

Note If time zone information changed due to periodic updates and the campaign's configured time zone is no longer valid, the following information is returned:

```
<timeZone>
   <refURL>/unifiedconfig/config/timezone/INVALID</refURL>
   <displayName>INVALID Time Zone</displayName>
</timeZone>
```

- startTime: The time the campaign starts dialing customer numbers. The format for this parameter is hours:minutes. Range is from 00:00 to 23:59. The default value is taken from the Blended_Agent_Options table column values for DialStartHours and DialStartMinutes.
- endTime: The time the campaign stops dialing customer numbers. The format for this parameter is hours:minutes. Range is from 00:00 to 23:59. The default value is taken from the Blended_Agent_Options table column values for DialEndHours and DialEndMinutes.
- enabled: Whether the dialer is available to call contacts. True or false. Default is false.
- startDate: The date that the campaign starts. The format is YYYY-MM-DD.
- endDate: The date that the campaign ends. The format is YYYY-MM-DD.
- campaignPrefix: Digits to prefix to each customer number dialed from this campaign. Maximum length of 15 digits.
- dialingMode: The dialing mode to use for the campaign skill groups. Valid values are as follows:
  - INBOUND
  - PREDICTIVEONLY
  - PREVIEWONLY
  - PROGRESSIVEONLY
  - PREVIEWDIRECTONLY
- reservationPercentage: The percentage of agents to reserve within the skill groups associated with the campaign. Range is from 0 to 100. Default is 100.
- callProgressAnalysis: A collection of parameters for Call Progress Analysis (CPA). Any combination of parameters within this collection can be set. If none of the parameters are provided and CPA is enabled for the campaign by default, CPA recording is set to false and default parameter values are set from the Blended_Agent_Options table.
  - enabled: When set to false, CPA for all calls made from this Dialer is disabled on a campaign-by-campaign basis, including voice detection, fax/modem detection, and answering machine detection. True or false. Default is true.
  - record: If enabled is set to true, you can specify this parameter. If you set it to true, the gateway provides a media stream and the Dialer records .wav files. True or false. Default is false.
  - minSilencePeriod: The minimum silence period (in milliseconds) required to classify a call as voice detected. If many answering machine calls are being passed through to agents as voice, then increasing this value accounts for longer pauses in answering machine greetings. Range is from 100 to 1000. Default is 608.
  - analysisPeriod: The number of milliseconds spent analyzing this call. If there is a short agent greeting on an answering machine, then a longer value categorizes that answering machine call as voice. If the call is to a business where the operator has a longer scripted greeting, a shorter value categorizes the long, live greeting as an answering machine call. Range is from 1000 to 10000. Default is 2500.
  - minimumValidSpeech: Minimum number of milliseconds of voice required to classify a call as voice detected. Range is 50 to 500. Default is 112.

- maxTimeAnalysis: The maximum number of milliseconds allowed for analysis before identifying a problem analysis as dead air/low volume. Range is 1000 to 10000. Default is 3000.
- maxTermToneAnalysis: The maximum number of milliseconds the dialer analyzes an answering machine voice message looking for a termination tone. If the message has an odd tone and the analysis does not recognize it, the call is not transferred or dropped until this timeout occurs. Range is 1000 to 60000. Default is 30000.

- skillGroupInfos: A collection of information about the skill groups associated with the campaign.
  - skillGroupInfo: A collection of information about one skill group.
    - skillGroup: The name and the refURL of the skill group assigned to the campaign.
    - overflowAgents: This parameter ensures that at least one extra agent is reserved before the dialer begins dialing for a Progressive campaign. If the parameter is set to 1, at least two agents must be reserved before the dialer begins dialing. Range is 0 to 100. Default is 0.
    - dialedNumber: The digits that are dialed to reserve an agent in the configured skill group. This parameter can contain letters, numbers, periods (.), and underscores (_) and can be up to ten characters in length.
    - recordsToCache: The minimum number of dialing numbers that each dialer caches for each of the Outbound Option skill groups. Range is 1 to 400. Default is 1.
    - ivrPorts: The total number of IVR ports allocated for the skill group. This parameter indicates how many ports are available for the dialer to transfer customer calls. Because this value indicates the total number of ports supported by the IVR for the current skill group, multiple skill groups can make transfer to IVR calls. One IVR can also be used to play different messages based on the route point where the contact is transferred. If multiple dialers are associated with the skill group, each dialer dials a fraction of the total number of ports. Default is 0.
    - ivrRoutePoint: If the campaign is a Transfer to IVR campaign or is configured to transfer AMD calls to an IVR, this parameter indicates the route point required to run the transfer to IVR routing script. This parameter must coincide with a route point configured on Unified Communications Manager and be assigned to a PGUser. Contacts are transferred to the route point, which points to a routing script. The script transfers the call to an IVR. Maximum length of 32 characters.
    - abandonedRoutePoint: If the campaign is a Transfer to IVR campaign or is configured to transfer AMD calls to an IVR, this number allows the dialer to play a message to calls about to be disconnected because no agents are available. This number must coincide with a route point configured on Unified Communications Manager and be assigned to the agent PG's CTI application (for example, PGUser). Contacts are transferred to this route point, which points to a routing script. The script transfers the call to an IVR. Maximum length of ten characters.

- skillGroupInfosAdded: A collection of skill group references to be added to the campaign, including the skill group refURL. This parameter is update only. This parameter can be used with the skillGroupInfosRemoved parameter. See References, on page 5.
- skillGroupInfosRemoved: A collection of skill group references to be removed from the campaign, including the skill group refURL of each skill group. This parameter is update only. This parameter can be used with the skillGroupInfosAdded parameter. See References, on page 5.

Search and Sort Parameters

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description • startDate • endDate • enabled • campaignPurposeType • timezone • dialingMode

See Search, on page 10 and Sort, on page 11.

Disable All Campaigns (Emergency Stop)

The Outbound Campaign API provides a method to immediately disable all outbound campaigns that are currently enabled. To disable all of the campaigns, perform a PUT on the following URL:

https://<server>/unifiedconfig/config/campaign/disable

This request retrieves a list of all campaigns created using the Outbound Campaign API that are currently enabled and sets the enabled parameter to false. The Campaign Manager stops sending out records to the Dialer for these campaigns. All active records in the Dialer's memory are removed.

Example Get Response

```
<campaign>
    <changeStamp>48</changeStamp>
    <refURL>/unifiedconfig/config/campaign/5168</refURL>
    <import>
        <refURL>/unifiedconfig/config/import/5000</refURL>
    </import>
    <abandonEnabled>true</abandonEnabled>
    <abandonPercent>3.0</abandonPercent>
    <amdTreatmentMode>abandonCall</amdTreatmentMode>
    <campaignPrefix>978</campaignPrefix>
    <campaignPurposeType>agentCampaign</campaignPurposeType>
    <dialingMode>PREVIEWONLY</dialingMode>
    <enabled>true</enabled>
    <endDate>2016-01-15</endDate>
    <endTime>17:00</endTime>
    <ipAMDEnabled>true</ipAmdEnabled>
    <ipTerminatingBeepDetect>false</ipTerminatingBeepDetect>
    <linesPerAgent>1.5</linesPerAgent>
    <maxAttempts>3</maxAttempts>
    <maximumLinesPerAgent>100.0</maximumLinesPerAgent>
    <minimumCallDuration>1</minimumCallDuration>
    <name>APIoct1</name>
    <noAnswerRingLimit>4</noAnswerRingLimit>
```

```
    <personalizedCallbackEnabled>false</personalizedCallbackEnabled>
    <predictiveCorrectionPace>70</predictiveCorrectionPace>
    <predictiveGain>1.0</predictiveGain>
    <rescheduleCallbackMode>useCampaignDN</rescheduleCallbackMode>
    <reservationPercentage>100</reservationPercentage>
    <retries>
        <answeringMachineDelay>60</answeringMachineDelay>
        <busySignalDelay>60</busySignalDelay>
        <customerAbandonedDelay>30</customerAbandonedDelay>
        <customerNotHomeDelay></customerNotHomeDelay>
        <dialerAbandonedDelay>60</dialerAbandonedDelay>
        <noAnswerDelay>60</noAnswerDelay>
    </retries>
    <skillGroupInfos>
    <skillGroupInfo>
        <ivrPorts>0</ivrPorts>
        <overflowAgents>0</overflowAgents>
        <recordsToCache>1</recordsToCache>
        <abandonedRoutePoint>12345</abandonedRoutePoint>
        <dialedNumber>123</dialedNumber>
        <ivrRoutePoint>91234</ivrRoutePoint>
        <skillGroup>
            <refURL>/unifiedconfig/config/skillgroup/5004</refURL>
            <name>errorDetailsRouteScript</name>
        </skillGroup>
    </skillGroupInfo>
    </skillGroupInfos>
    <startDate>2016-01-14</startDate>
    <startTime>09:00</startTime>
    <timeZone>
        <displayName>(UTC-05:00) Eastern Time (US & Canada)</displayName>
        <refURL>/unifiedconfig/config/timezone/Eastern%20Standard%20Time</refURL>

    </timeZone>
    <callProgressAnalysis>
        <enabled>true</enabled>
        <record>false</record>
        <minSilencePeriod>608</minSilencePeriod>
        <analysisPeriod>2500</analysisPeriod>
        <minimumValidSpeech>112</minimumValidSpeech>
        <maxTimeAnalysis>3000</maxTimeAnalysis>
        <maxTermToneAnalysis>30000</maxTermToneAnalysis>
    </callProgressAnalysis>
</campaign>
```

Example Create Request

```
<campaign>
    <name>APIOct1</name>
    <description>APIOct1</description>
    <dialingMode>PREVIEWONLY</dialingMode>
    <skillGroupInfos>
        <skillGroupInfo>
            <ivrPorts>0</ivrPorts>
            <overflowAgents>0</overflowAgents>
            <recordsToCache>1</recordsToCache>
            <skillGroup>
                <refURL>/unifiedconfig/config/skillgroup/5001</refURL>
                <name>sgcampaign</name>
            </skillGroup>
        </skillGroupInfo>
    </skillGroupInfos>
    <timeZone>
```

```
        <refURL>/unifiedconfig/config/timezone/UTC</refURL>
    </timeZone>
    <callProgressAnalysis>
        <enabled>true</enabled>
        <record>false</record>
        <minSilencePeriod>608</minSilencePeriod>
        <analysisPeriod>2500</analysisPeriod>
        <minimumValidSpeech>112</minimumValidSpeech>
        <maxTimeAnalysis>3000</maxTimeAnalysis>
        <maxTermToneAnalysis>30000</maxTermToneAnalysis>
    </callProgressAnalysis>
</campaign>
```

Campaign Status API

Use the Campaign Status API to get the real-time status of running Outbound Option campaigns.

URL

```
https://<server>/unifiedconfig/config/campaign/<campaign-id>/runtimestats
```

Note You must specify the ID for a campaign that was created using the Outbound Campaign API (see Outbound Campaign API, on page 154). If you specify the ID for a campaign created using the Outbound Option Campaign tool, the request returns CceDBDataNotFoundException.

Operations

- get: Returns the runtime status of a campaign.

Response Parameters

- abandonDetectCount: The number of calls abandoned by the dialer.
- abandonToIvrCount: The number of calls that were abandoned by the dialer and transferred to IVR.
- agentClosedCount: The number of preview and callback calls that were closed by the agent.
- agentRejectedCount: The number of preview and callback calls that were rejected by the agent.
- answeringMachineCount: The number of calls that detected an answering machine.
- attemptedCount: Summary total of the number of calls attempted.
- busyCount: The number of calls that detected a busy signal.
- callBackCount: The number of callback contacts.
- cancelledDetectCount: The number of calls where the dialer canceled a ringing customer call.
- closedCount: The number of contacts closed for any reason other than reaching a live customer.
- customerAbandonDetectCount: The number of calls where the customer hung up immediately after picking up the phone.

- customerNotHomeCount: The number of contacts where the party who answered the phone was not the customer.
- dateTime: The date and time when this data was updated last. The format is yyyy-MM-ddTHH:mm:ss (for example, 2016-03-13T04:50:31).
- faxDetectCount: The number of calls that detected a fax machine.
- networkAnsMachineCount: The number of calls that detected a network answering machine.
- noAnswerDetectCount: The number of calls that were not answered.
- noDialToneDetectCount: The number of calls that did not detect a dial tone.
- noRingBackDetectCount: The number of calls that did not detect a ring back.
- personalCallbackCount: The number of callback contacts scheduled.
- sitToneDetectCount: The number of calls that detected a special information tone (SIT).
- talkTimeCount: The total number of seconds that agents spent talking on the phone since midnight.
- totalCount: The total number of records available to dial for the current campaign.
- totalVoiceCount: The number of live customers that have been reached for this campaign since the last time the imported dialing list was overwritten.
- voiceCount: The number of calls for the day that ended in successful customer contact.
- wrapupTimeCount: The number of seconds that agents spent in wrap-up mode since midnight.
- wrongNumberCount: The number of contacts where the party who answered the phone indicated that the customer did not live there.

Example Get Response

```
<runtimeStatus>
    <abandonDetectCount>0</abandonDetectCount>
    <abandonToIvrCount>0</abandonToIvrCount>
    <agentClosedCount>0</agentClosedCount>
    <agentRejectedCount>0</agentRejectedCount>
    <answeringMachineCount>0</answeringMachineCount>
    <attemptedCount>1</attemptedCount>
    <busyCount>0</busyCount>
    <callBackCount>0<callBackCount>
    <cancelledDetectCount>0</cancelledDetectCount>
    <closedCount>0</closedCount>
    <customerAbandonDetectCount>0</customerAbandonDetectCount>
    <customerNotHomeCount>0</customerNotHomeCount>
    <dateTime>2016-01-15T13:43:00</dateTime>
    <faxDetectCount>0</faxDetectCount>
    <networkAnsMachineCount>0</networkAnsMachineCount>
    <noAnswerDetectCount>0</noAnswerDetectCount>
    <noDialToneDetectCount>0</noDialToneDetectCount>
    <noRingBackDetectCount>0</noRingBackDetectCount>
    <personalCallbackCount>0</personalCallbackCount>
    <sitToneDetectCount>0</sitToneDetectCount>
    <talkTimeCount>1</talkTimeCount>
    <totalCount>0</totalCount>
    <totalVoiceCount>1</totalVoiceCount>
    <voiceCount>1</voiceCount>
```

```
    <wrapupTimeCount>0</wrapupTimeCount>
    <wrongNumberCount>0</wrongNumberCount>
</runtimeStatus>
```

Do Not Call API

Use the Do Not Call API to set the Do Not Call (DNC) import rule configuration for Outbound Option so that the campaign dialer doesn't dial the numbers in the DNC list.

The DNC import is automatically scheduled to start when the file is available. After the import is complete, the file is RENAMED or DELETED based on the boolean value for the renameFileAfterImport field.

URL

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

Operations

- create: Sets the configuration in the database. Returns the refURL if the configuration is set.
- delete: Deletes the specific DNC configuration using the URL `https://<server>/unifiedconfig/config/dnc/<id>`. This does not delete the DNC import file that is present at the location of the filePath.
- list: Retrieves a list available DNC import rules.
- get: Returns the DNC import configuration from the database using the URL `https://<server>/unifiedconfig/config/dnc/<id>`.
- update: Updates the specific DNC import configuration with the new values using the URL `https://<server>/unifiedconfig/config/dnc/<id>`.

Parameters

- name: Name of the DNC import. See Shared Parameters, on page 8.
- filePath: Path to Microsoft Windows Distributed File System (DFS) share on the logger or accessible from the logger. Maximum length of 255 characters.
- overwrite: Whether the new DNC import overwrites the phone numbers from the previous DNC import. True or false. Default is false.
- renameFileAfterImport: Whether to delete the DNC import file after the import is complete. True or false. Default is true. If the parameter is set to true, the DNC file is renamed. If the parameter is set to false, the DNC import file is deleted.

Search and Sort Parameters

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name	• name

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<dnc>
    <refURL>/unifiedconfig/config/dnc/unifiedconfig/config/dnc/5001</refURL>
    <name>dnc1</name>
    <filePath>\\CCE-ROGGER-A\C$\Users\boston\Desktop\dnc.txt</filePath>
    <overwrite>false</overwrite>
    <renameFileAfterImport>true</renameFileAfterImport>
</dnc>
```

Create Do Not Call List

When creating a Do_Not_Call list file, format it correctly using the following instructions.

Procedure

Step 1 Using a text editor, create a text file that contains all the do-not-call phone numbers.

Step 2 Enter a phone number for each Do Not Call entry on a new line.

Step 3 Observe the following characteristics for each Do Not Call entry:

- Each phone number can be a maximum of 20 characters long.
- The Do Not Call table can support up to 10 million entries, but note that the information is also stored in memory in the Campaign Manager process. Unless you set the **Overwrite table** option, each import appends to the table. A single large import or several smaller ones can create a Do Not Call list in memory that consumes all the memory for the process.

Step 4 Save the text file to the local server.

The following is an example of a Do_Not_Call list:

```
2225554444

2225556666

2225559999
```

To add a customer to this list, import a Do Not Call list.

To save file path of Do Not Call list import file:

1. In Unified CCE Administration, choose **Organization** > **Campaigns** to open the **Campaigns** page.
2. Click the **Do Not Call - Settings** link.

   The **Do Not Call - Settings** pop-up window appears.
3. In the **File Path with Name** field, you must enter the DNC list import file path on the logger or the path accessible from the logger.
4. Click **Save**.

The solution import the DNC phone numbers to Do_Not_Call table in BA database. The name of DNC list import file is renamed after the successful import.

Note On the **Campaigns** page, you can save only one DNC list import file path at a time.

The campaign validates that a number in the dialing list is not in the Do Not Call list before sending it to a dialer. The solution checks the list at the last minute before placing the call. You can update a Do Not Call list while a campaign is running.

The Campaign Manager reads from the Do_Not_Call table. Dialing List entries are marked as Do Not Call entries only when the Campaign Manager fetches the Dialing List entry *and only when there is an exact, digit-for-digit match*. This allows Do Not Call imports to happen while a Campaign is running without rebuilding the Dialing List.

Note If the Dialing List includes a base number plus extension, this entry must match a Do Not Call entry for that same base number and same extension. The dialer will not dial the extension.

To edit the DNC phone numbers import file path:

1. Click the **Do Not Call - Settings** link. The solution displays the existing file path of the DNC list import file in the **File Path with Name** field
2. Enter the file path of the new or updated DNC list import file in the **File Path with Name** field.
3. (Optional) Check the **Delete existing "Do Not Call" Phone Numbers** check box to delete the existing phone numbers from the Do_Not_Call table.

   If this check box is not selected, the existing DNC phone numbers are appended to the new or updated DNC phone numbers in the Do_Not_Call table.
4. Click **Save**.

Note To clear the Do Not Call list, import a blank file with the Overwrite table option enabled.

Import API

Use the Import API to import customer contact information for an outbound campaign.

You can import up to 10,000 records in one create request.

Note You can queue up to 30 requests at a time. If you have 30 requests in queue and submit another request, the response is HTTP 503 Service Unavailable with the boundary error condition "Request processing queue is at capacity."

If you receive this error, wait until a few requests are processed and the queue goes below 30, then submit the next request.

If a particular integration is doing multiple small inserts per second for a campaign and sees this error, it is generally more efficient to batch up the records into larger imports.

URL

`https://<server>/unifiedconfig/config/campaign/<id>/import`

Note You must specify the ID for a campaign that was created using the Outbound Campaign API. If you specify the ID for a campaign created using the Outbound Option Campaign tool, the request returns CceDBDataNotFoundException.

Operations

- create: Imports customer contacts for a specific campaign.
- delete: Deletes the imported contacts for a specific campaign.

  Note The delete operation deletes all imported contacts for the campaign. You cannot delete an individual record.

- get (template): Returns a sample CSV template for contacts, which is provided by the API, using the URL `https://<server>/unifiedconfig/config/campaign/import/template`. The response contains the CSV template as a file attachment.
- get: Retrieves the details of a single import record using the URL https://<server>/unifiedconfig/config/campaign/<campaignId>/import/<id>. The <id> is the ID of the imported contact.
- get (csv file): Returns a CSV file for the contacts of the campaign, which is provided by the API, using the URL `https://<server>/unifiedconfig/config/campaign/<campaignId>/import` with header as "Accept" and value as "text/csv". The response contains the `Contacts.csv` as a file attachment.
- get (csv file): Returns a CSV file for the contacts of the campaign, which is provided by the API, using the URL `https://<server>/unifiedconfig/config/campaign/<campaignId>/import/csv`. The response contains the `Contacts.csv` as a file attachment. In this operation, you do not need to send "Accept" header.

- list: Retrieves a list of all imported contacts for the campaign.
  - **Query parameters**
    - Summary list: See list, on page 3.

Parameters

Response Parameters for get:

- accountNumber: The customer's account number.
- callsMade: The number of calls made.
- callStatus: The call status of the last call placed for this record. Possible values include the following:
  - active (A)
  - callbackRequested (B)
  - closed (C)
  - dialed (D)
  - agentRejected (J)
  - maxAttemptsReached (M)
  - pending (P)
  - retry (R)
  - personalCallbackRequested (S)
  - unknown (U)

  For more information about these values, see the CallStatusZone Values section of the *Outbound Option Guide for Unified Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html.
- callResultOverall: The call result from the last call placed for this record.
- firstName: The customer's first name.
- lastName: The customer's last name.
- importDate: The date and time that the record was imported.
- phone01 through phone10: A collection of information about the customer's phones. Each phone contains the following parameters:
  - number: The phone number.
  - callResult: The call result from the last call placed for this phone number.

    For more information about possible callResult values, see the CallResult Codes and Values section of the *Outbound Option Guide for Unified Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html.
  - dstObserved: Whether DST is observed for this phone's location. True or false.

- gmtOffset: The number of minutes that the customer's time zone is offset from GMT. When you create import records, you can optionally provide a TimeZoneBias. The TimeZoneBias parameter represents the way information about time zones is stored in the Windows registry. You can use the following formula to convert TimeZoneBias to gmtOffset:

```
if timeZoneBias is postive:
    gmtOffset = 1440 - timeZoneBias (where 1440 is the number of minutes in 24
hours)
else if timeZoneBias is negative:
    gmtOffset = -1 * timeZoneBias
else if timeZoneBias is 0
    gmtOffset = timeZoneBias
```

Parameters for create:

- fileContent: Required. Comma-separated or pipe-separated list of data embedded within a CDATA section.
- overwriteData: True or false. If set to true, the existing import data in the database is overwritten by the new import data. If set to false, the new import data is appended to the existing data. Default is false.

Note You cannot modify or remove existing fields or append a new field to existing data.

- delimiter: Comma (,) or pipe (|). Default is comma (,).
- header fields: The following header fields are allowed:
  - AccountNumber: Can contain any characters, including internationalized characters, except the delimiter specified in the input XML. Maximum length of 30 bytes.
  - FirstName: Can contain any characters, including internationalized characters, except the delimiter specified in the input XML. Maximum length of 50 characters.
  - LastName: Can contain any characters, including internationalized characters, except the delimiter specified in the input XML. Maximum length of 50 characters.
  - Phone01 through Phone10: Can contain digits 0-9, pound sign (#), and asterisk (*). Maximum length of 20 bytes. At least one Phone field is required.
  - TimeZoneBias: Specifies the offset of the customer's time zone from UTC in minutes. Integer. Range is from –780 to 720 (–13 to 12 hours from UTC).
  - DstObserved: Specifies whether DST is observed for the customer's location. True or false. Default is false.

Note If the TimeZoneBias parameter is provided but the DstObserved parameter is not, DstObserved is set to false. If the DstObserved parameter is provided but not TimeZoneBias, a validation error is returned.

If TimeZoneBias is not provided, time zone and Daylight Saving Time information is assigned by matching phone numbers to region prefix strings. If a phone number for a contact does not match a configured region prefix, the default time zone for the campaign is used.

If a customer record contains multiple phone numbers, the value for TimeZoneBias and DstObserved are applied to all of that customer's phone numbers.

You can use the Time Zone API to look up the values for TimeZoneBias and DstObserved for the target time zone. For more information, see Time Zone API, on page 175.

Search and Sort

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• accountNumber • firstName • lastName	• accountNumber (default) • firstName • lastName

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<importContact>
    <refURL>/unifiedconfig/config/campaign/5000/import/1</refURL>
    <accountNumber>4019</accountNumber>
    <callsMade>1</callsMade>
    <callStatus>pending</callStatus>
    <callResultOverall>0</callResultOverall>
    <firstName>Mir</firstName>
    <lastName>Ali</lastName>
    <importDate>2016-03-28T01:09:40</importDate>
    <phone01>
        <callResult>0</callResult>
        <dstObserved>true</dstObserved>
        <gmtOffset>720</gmtOffset>
        <number>9789360001</number>
    </phone01>
    <phone02>
        <callResult>0</callResult>
        <dstObserved>true</dstObserved>
        <gmtOffset>720</gmtOffset>
        <number>9789360002</number>
    </phone02>
    <phone03>...</phone03>
```

```
    <phone04>...</phone04>
    <phone05>...</phone05>
    <phone06>...</phone06>
    <phone07>...</phone07>
    <phone08>...</phone08>
    <phone09>...</phone09>
    <phone10>
        <callResult>0</callResult>
        <dstObserved>true</dstObserved>
        <gmtOffset>330</gmtOffset>
        <number>9789360010</number>
    </phone10>
</importContact>
```

Example Create Request

```
<import>
    <fileContent>
        <![CDATA[
        AccountNumber,FirstName,LastName,Phone01,Phone02,TimeZoneBias,DstObserved
        6782,Henry,Martin,2225554444,2225556262,720,false
        3456,Michele,Smith,2225559999,2225551234,540,true
        4569,Walker,Evans,2225552000,2225557890,-600,true
        ]]>
    </fileContent>
    <delimiter>,</delimiter>
</import>
```

Personal Callback API

Use the Personal Callback API to configure your Outbound Option campaign to handle personal callbacks. The Personal Callback feature allows an agent to schedule a callback to a customer for a specific date and time.

URL

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

Operations

- create: Creates a single PersonalCallback record and stores it in the BA database.
- create (bulk): Creates PersonalCallback records in bulk and stores them in the BA database using the URL `https://<server>/unifiedconfig/config/personalcallback/import`.
- delete: Deletes one or more PersonalCallback records from the PersonalCallback list using the URL `https://<server>/unifiedconfig/config/personalcallback/<id>`.
- get (template): Retrieves a sample CSV template for contacts, which is provided by the API, using the URL `https://<server>/unifiedconfig/config/personalcallback/template`.
- get (record): Retrieves one PersonalCallback record from the database using the URL `https://<server>/unifiedconfig/config/personalcallback/<id>`.
- list: Retrieves a list of PersonalCallback records from the database.
    - **Query parameters:**

- agentId
- accountNumber
- firstName
- lastName

- update: Updates one PersonalCallback record in the database using the URL `https://<server>/unifiedconfig/config/personalcallback/<id>`.

Parameters

Parameters for create (single record), get, and update operations:

- campaign: A reference to a specific campaign, including the refURL and campaign ID.
- agent: A reference to a specific agent, including the refURL and agent ID.
- campaignId: Read only. The ID for the campaign.
- peripheralId: Read only. ID for the peripheral on which the agent would be available.
- agentId: Read only. ID of the agent to whom to connect the call.
- campaignDn: The Dialed Number (DN) to use if the original agent is not available.
- phone: Required. The phone number to call back. Can contain digits 0 to 9, pound sign (3), and asterisk (*). Maximum length of 20 bytes.
- accountNumber: The customer's account number. Internationalized characters are allowed. Maximum length of 30 bytes.
- maxAttempts: Required. The maximum number of times to attempt a call (decrements at each attempt). An attempt is the Dialer's attempt to reserve the agent and call the customer. Because the Dialer places multiple customer call attempts (such as busy, no answer), individual call attempts are not tracked here; only the result at the end of the callback time range. After this parameter is set to 0, no more attempts are made. Must be a positive integer value.
- callbackDateTime: Required. The time to attempt the customer callback is normalized to the logger GMT zone. For example, the Campaign Manager is in Boston and the customer is in California. The customer wants to be called back at 3 p.m. The time in this column is 6 p.m. The format for this parameter is yyyy-MM-ddTHH:mm:ss (for example, 2016-03-13T04:50:31).

  The following rules apply to the callbackDateTime parameter:

  - The callbackDateTime that is used is local to the time zone of the Logger. You need to consider this if the client machine on which you are using this API is in a different time zone than the Logger machine.
  - The setting of the callbackDateTime by this API reads the CallbackDateTimeLimit in the registry. For more information, see the Registry Settings section of the *Outbound Option Guide for Unified Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html.
  - The Campaign Manager uses the CallbackDateTimeLimit registry value to select Personal Callback records to queue to the Dialer.

- The default CallbackDateTimeLimit in the registry is 15 minutes. For create and update operations, you cannot set the callbackDateTime to 15 minutes or more before the current time. For example, if it is currently 4:00, you cannot set the callbackDateTime to 3:30.
- For update operations, the new callbackDateTime value can only be set if the current callbackDateTime is 15 minutes or more after the current Logger time. For example, if the current Logger time is 4:00 and the current callbackDateTime is 4:30, the new callbackDateTime can be set to 4:10. However, if the current callbackDateTime is 4:10, it cannot be set to 4:00.

- callStatus: The status of the personal callback. Possible values include the following:
  - active (A)
  - callbackRequested (B)
  - closed (C)
  - dialed (D)
  - agentRejected (J)
  - maxAttemptsReached (M)
  - pending (P)
  - retry (R)
  - personalCallbackRequested (S)
  - unknown (U)
  - agentNotAvailable (X)

  For more information about these values, see the CallStatusZone Values section of the *Outbound Option Guide for Unified Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html.
- callResult: Read only. Corresponds to the CallResult value in the Dialer_Detail table.

  For more information about possible callResult values, see the CallResult Codes and Values section of the *Outbound Option Guide for Unified Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html.
- lastName: The customer's last name. Internationalized characters are allowed. Maximum length of 50 characters.
- firstName: The customer's first name. Internationalized characters are allowed. Maximum length of 50 characters.

Bulk parameters:

- fileContent: Comma-separated or pipe-separated list of data embedded within a CDATA section.
- overwriteData: True or false. If set to true, the existing personal callback data in the database is overwritten by the new personal callback data. If set to false, the new personal callback data is appended to the existing data. Default is false.

Note You cannot modify or remove existing fields or append a new field to existing data.

If overwriteData is set to true, any agent scheduled personal callback records are also overwritten.

- delimiter: Comma (,) or pipe (|). Default is comma (,).

Bulk fileContent parameters:

Note The descriptions for these parameters are the same as the descriptions for the parameters for create (single record), get, and update operations. Only the case is different.

- CampaignId (must be a valid campaign ID in the database)
- AgentSkillTargetId (must be a valid SkillTargetID in the database)
- CampaignDn
- Phone
- AccountNumber
- MaxAttempts
- CallbackDateTime
- LastName
- FirstName

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• agentId • accountNumber • firstName • lastName	• agentId • accountNumber • firstName • lastName

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

There are several advanced searches you can perform on the Personal Callback API, including agentId, accountNumber, firstName, and lastName. All search terms are case-insensitive.

- **agentId:<ID>** finds all personal callback records where the agent ID contains a specified ID. For example, **agentId:123** returns all records where the agent ID contains the string "123".

- **lastName:<name>** finds all personal callback records where the last name contains the specified name. For example, **lastName:smith** returns all records where the last name includes "smith".
- **firstName:<name>** finds all personal callback records where the firstName contains the specified name. For example, **firstName:John** returns all records where the first name includes "john".
- **accountNumber:<number>** finds all personal callback records where the accountNumber contains the specified number. For example, **accountNumber:456** returns all records where the account number includes "456".

Example Create Request (Single Record)

```
<personalCallback>
    <campaign>
        <refURL>/unifiedconfig/config/campaign/5000</refURL>
    </campaign>
    <agent>
        <refURL>/unifiedconfig/config/agent/5050</refURL>
    </agent>
    <campaignDn>222222</campaignDn>
    <phone>999333</phone>
    <accountNumber>23334343334</accountNumber>
    <maxAttempts>1</maxAttempts>
    <callbackDateTime>2016-01-15T11:37:00</callbackDateTime>
    <callStatus>pending</callStatus>
    <lastName>Kumar</lastName>
    <firstName>Akshaya</firstName>
</personalCallback>
```

Example Create Request (Bulk)

```
<personalCallback>
    <fileContent>
        <![CDATA[
        AccountNumber,FirstName,LastName,Phone,AgentSkillTargetId,CampaignId,CampaignDn,
        CallbackDateTime,MaxAttempts
        6782,Henry,Martin,2225554444,1004,5000,2222222221,2016-01-25T05:12:00,1
        3444,Thomas,Edison,2225554555,1004,5000,2222222222,2016-01-25T05:23:00,1
        5444,Tom,Hilfiger,2225554666,1004,5000,2222222223,2016-01-25T05:37:00,1
        ]]>
    </fileContent>
</personalCallback>
```

Example Get (Record) Response

```
<personalCallback>
    <changeStamp>48</changeStamp>
    <refURL>/unifiedconfig/config/campaign/personalcallback/2</refURL>
    <campaign>
        <refURL>/unifiedconfig/config/campaign/5000</refURL>
    </campaign>
    <agent>
        <refURL>/unifiedconfig/config/agent/5050</refURL>
    </agent>
    <campaignId>5000</campaignId>
    <peripheralId>5000</peripheralId>
    <agentId>1001</agentId>
    <campaignDn>222222</campaignDn>
```

```
    <phone>999333</phone>
    <accountNumber>23334343334</accountNumber>
    <maxAttempts>1</maxAttempts>
    <callbackDateTime>2016-01-15T11:37:00</callbackDateTime>
    <callStatus>pending</callStatus>
    <callResult>0</callResult>
    <lastName>Kumar</lastName>
    <firstName>Akshaya</firstName>
</personalCallback>
```

Time Zone API

Use the Time Zone API to list all available time zones and to get time zone information for a specified zone. Time zone information is stored in the registry of the Windows operating system.

Important Microsoft periodically releases cumulative time zone updates. These updates include worldwide changes to time zone names, bias (the amount of time in minutes that a time zone is offset from Coordinated Universal Time (UTC)), and observance of daylight saving time. These patches update the information in the Windows registry. When these updates are available, apply them to all virtual machines in the deployment that are running a Microsoft Windows operating system.

Use this API with the Outbound Campaign API to set the default time zone for an Outbound Option campaign. An Outbound Option campaign uses the time zone when the location of the customer number being dialed is unknown.

This API is read-only.

URL

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

Operations

- list: Retrieves a list of available time zones. The list is sorted by UTC offset from the International Date Line from west to east.
- get: Returns information for a specific time zone using the URL `https://<server>/unifiedconfig/config/timezone/<id>`, where <id> is the URL-encoded version of the name parameter.

Response Parameters

- name: The name of the time zone.
- displayName: Specific bias and location information about the time zone, such as the offset from UTC and one or more places located within the time zone.

  Example: “(UTC+5:30) Chennai, Kolkata, Mumbai, New Delhi”
- stdName: The time zone name during standard time.

  Example: Malay Peninsula Standard Time

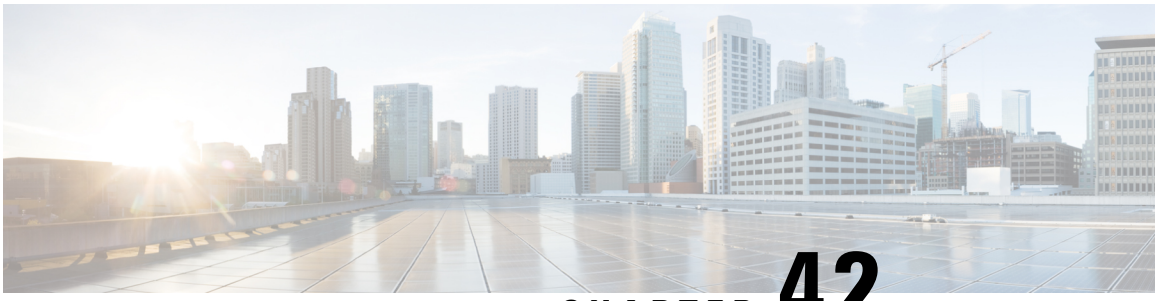
- dstName: The time zone name during daylight saving time.

  Example: Malay Peninsula Daylight Time

- dstObserved: Indicates whether daylight saving time is observed for the time zone. True or false.
- bias: The current bias for local time translation on the server (in minutes). That is, the number of minutes to add to the local time to yield UTC.

Example Get Response

```
<timeZone>
   <refURL>
     /unifiedconfig/config/timezone/Pacific%20Standard%20Time%20%28Mexico%29
   </refURL>
   <name>Pacific Standard Time (Mexico)</name>
   <displayName>(UTC-08:00) Baja California</displayName>
   <stdName>Pacific Standard Time (Mexico)</stdName>
   <dstName>Pacific Daylight Time (Mexico)</dstName>
   <dstObserved>true</dstObserved>
   <bias>480</bias>
</timeZone>
```



CHAPTER 42

Peripheral Gateway API

Use the Peripheral Gateway (PG) API to retrieve peripheral gateway information.

URL

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

Operations

- list: Retrieves a list of peripheral gateways.
- get: Returns one peripheral gateway using the URL `https://<server>/unifiedconfig/config/peripheralgateway/<id>`.

Parameters

- refURL: The refURL of the peripheral gateway. See Shared Parameters, on page 8.
- name: The name of the peripheral gateway. See Shared Parameters, on page 8.
- logicalControllerId: The ID of the logical controller.
- peripherals: A collection of peripheral information, including client type, name, peripheralID, routingClientID, and routingType (see Dialed Number API, on page 95 for routingType values).
    - The client type values are:
        - 13: VRU
        - 30: CUCM
        - 42: Generic PG
        - 47: MediaRouting
- datacenter: A reference to the data center, including the refURL and name.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search Parameters	Sort Parameters
name	• name (Default) • datacenter.name

See Search, on page 10 and Sort, on page 11.

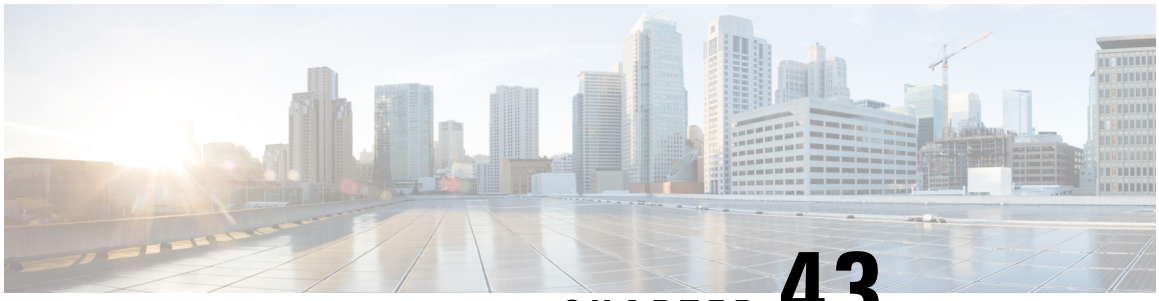
Advanced search parameters

The Peripheral Gateway API has an advanced search for datacenters.

- **datacenters: (dc1|dc2|dc3...)** which returns all peripheral gateways which belong to any of the specified data centers. You can specify up to three data centers. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all machines in the core data center.

Example Get Response

```
<peripheralGateway xsi:type="peripheralGateway">
     <changeStamp>0</changeStamp>
     <refURL>/unifiedconfig/config/peripheralgateway/5001</refURL>
     <name>MR_PG</name>
     <logicalControllerID>5001</logicalControllerID>
<datacenter>
                <name>Berlin</name>
                <refURL>/unifiedconfig/config/datacenter/5000</refURL>
        </datacenter>
     <peripherals>
         <peripheral>
             <changeStamp>824</changeStamp>
             <clientType>47</clientType>
             <name>Multichannel</name>
             <peripheralID>5005</peripheralID>
             <routingClientID>5005</routingClientID>
             <routingType>3</routingType>
         </peripheral>
         <peripheral>
             <changeStamp>822</changeStamp>
             <clientType>47</clientType>
             <name>Outbound</name>
             <peripheralID>5007</peripheralID>
             <routingClientID>5007</routingClientID>
             <routingType>4</routingType>
         </peripheral>
     </peripherals>
</peripheralGateway>
```



CHAPTER 43

Precision Queue API

Precision queues help direct incoming callers to appropriate agents, as they match specific agent attributes with caller requirements. If a precision queue requires an agent who lives in Boston and who speaks fluent Spanish, then an agent with the attributes **Boston = True** and **Spanish = True** is a good match.

Use the Precision Queue API to list the precision queues currently defined in the database, define new precision queues, and view, edit, and delete existing precision queues.

URL

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

Operations

- create: Creates one precision queue.
- delete: Marks one precision queue for deletion, but does not permanently delete it. Deleting a precision queue that is referenced dynamically in a script is allowed. No new calls are queued against it, but the precision queue remains operational until calls are no longer in the queue.
- get: Returns one precision queue, using the URL `https://<server>/unifiedconfig/config/precisionqueue/<id>`.
    - **Query parameters:**
        - agentcount: Use this query parameter to have the agent count parameter included in the response; for example, agentcount=true.
        - attributes: Use this query parameter to have the attribute parameter included in the response; for example, attributes=true.
        - skillgroups: Use this query parameter to augment the returned precision queue attributes with an id listing of all of the skillgroups that are associated with the precision queue; for example, skillgroups=true.
- list: Retrieves a list of precision queues. Query parameters described above for the get operation are also allowed for list.
- update: Updates one precision queue.

Parameters

Precision queue parameters:

- refURL: The refURL of the precision queue. See Shared Parameters, on page 8.
- name: The name of the precision queue. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- department: A reference to the department (Department API, on page 89), including the refURL and name. See References, on page 5.
- mediaRoutingDomain: A reference to the media routing domain (Media Routing Domain API, on page 143) including the name and refURL. See References, on page 5.
  - Defaults to Cisco_Voice MRD if this parameter is not provided.
  - This reference cannot be updated.
- bucketInterval: A reference to a bucket interval (Bucket Interval API, on page 47), including the refURL and name. See References, on page 5.
- agentCount: Returns agent count for the precision queue. Returned only when using the agentcount query parameter.
- agentOrdering: Determines the order in which agents receive calls from this queue.
  - 1: LAA (Agent availability time)
  - 2: Most skilled agent
  - 3: Least skilled agent
- id: The database id of the precision queue. Read-only field. Used in scripting.
- attributes: A collection of attribute names (attribute1, attribute2, and so on) indicating all of the attributes used in this precision queue. Returned only when the query parameter attributes=true.
- 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.
  - 1: Ignore abandoned calls.
  - 2: Abandoned call has negative impact.
  - 3: Abandoned call has positive impact.
- skillgroups: A collection of skill groups associated with this precision queue, including the id of each skill group. Returned only when the query parameter skillgroups=true.
- steps: Required. A collection of steps for this precision queue. You can have 1-10 steps. Returned only for get operation. See the Step parameters below.

Step parameters:

- waitTime: Time in seconds to wait before proceeding to the next step.
- considerIf: A Consider If expression which must be met to execute a particular step. Items used in the expression are case sensitive. You cannot add an expression to the last step.
- terms: Required. A collection of terms for this step. Each step can have 1-10 terms. See the Term parameters below.

Term parameters:

- attribute: A reference to the attribute (Attribute API, on page 43), including the refURL, name, description, and dataType. Multiple unique attributes can be used across all terms in a precision queue.
- parenCount: Denotes a parenthesis before or after this term. A value of 1 means a parenthesis before the current term, and a value of -1 means a parenthesis after the current term. The sum of all parenCount for all terms in a step must be equal to zero, meaning that all parenthesis in the expression are matched. For example, a step to check for agents that have (sales > 7 or expertSales = true) and english = true requires 3 terms with the parenCount set to 1 on the first term, -1 on the second term, and 0 on the last term.
- termRelation: Indicates the relationship of this term to the preceding term, using the following values:
  - 0: None. Valid only on the first term in a step.
  - 1: AND
  - 2: OR
- attributeRelation: Indicates what kind of comparison is done on the attribute, using the following values:
  - 1: Equal
  - 2: Not equal
  - 3: Less than
  - 4: Less than or equal
  - 5: Greater than
  - 6: Greater than or equal
- value1: The value that the attribute is tested against. For boolean attributes, this value must be true/false. For proficiency attributes, this value must be 1-10.

Search and Sort Values

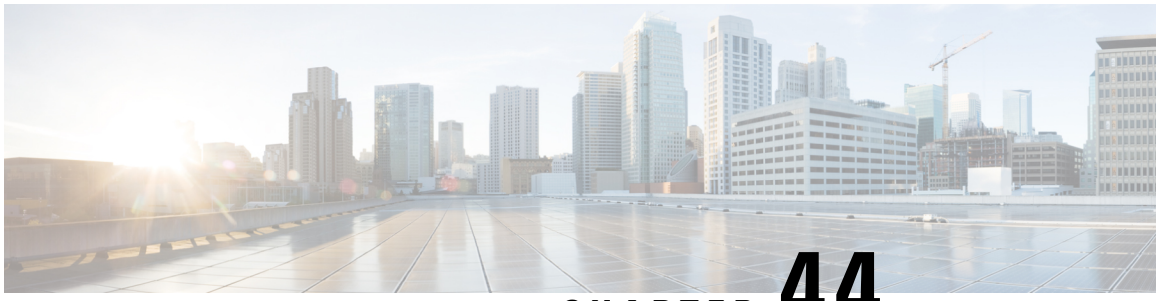
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<precisionQueue>

   <refURL>/unifiedconfig/config/precisionqueue/5002</refURL>
   <changeStamp>4</changeStamp>
   <agentOrdering>1</agentOrdering>
   <bucketInterval>
      <refURL>/unifiedconfig/config/bucketinterval/1</refURL>
      <name>Default_Bucket_Intervals</name>
   </bucketInterval>
   <mediaRoutingDomain>
      <name>Cisco_Voice</name>
      <refURL>/unifiedconfig/config/mediaroutingdomain/1</refURL>
   </mediaRoutingDomain>
   <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>
   </steps>
</precisionQueue>
```



CHAPTER 44

Reason Code API

The Reason Code feature in Packaged CCE is used to configure the Not Ready, Sign Out, and Wrap-Up reason codes.

Agents select the reason code on their agent desktops (Cisco Finesse) to provide the work status. Reason codes appear in the Unified Intelligence Center reports and help identify agent behavior.

You can configure Reason Codes in Packaged CCE, and the configured reason codes appear in the agent desktop (Cisco Finesse)

Use the Reason Code API to define new reason codes, and edit and delete records of the existing reason codes.

URL

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

Operations

- create: Creates one reason code.
- delete: Marks one reason code for deletion.
- get: Returns one reason code, using the URL `https://<server>/unifiedconfig/config/reasoncode/<id>`.
- list: Retrieves a list of reason codes.
- update: Updates one reason code.

Parameters

- refURL: The refURL of the reason code. See Shared Parameters, on page 8.
- changestamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- code: Required. The unique reason code. Integers between 1 and 65535. Value cannot be updated after the reason code has been created.

Note Do not specify code while configuring the Wrap-Up reasons.

- text: Required. The text that describes the reason code. Maximum length of 40 characters allowed.
- reasonType: The type of reason. Allows the following values - NOT_READY, WRAPUP, and LOGOUT.

  NOT_READY is the default reasonType. If reasonType is not specified in API, the system sets the reasonType as NOT_READY.
- isGlobal: The reason code must be global (parameter set to true) or be assigned to a team to which the user belongs (parameter set to false). If isGlobal is not specified in API, the system sets isGlobal to true.
- category: To identify the user-defined reason codes and system-defined (Not Ready and Sign Out) reason codes.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• text • description	• text (default) • description • code • reasonType • category

See Search, on page 10 and Sort, on page 11.

Example Get Response

```
<reasonCode>
   <changeStamp>0</changeStamp>
   <refURL>/unifiedconfig/config/reasoncode/5001</refURL>
   <code>1</code>
   <isGlobal>false</isGlobal>
   <description>This is a reason code.</description>
   <text>Reason Code1</text>
   <reasonType>WRAPUP</reasonType>
   <category>User-defined</category>
</reasonCode>
```

Maximum limit for Reason Codes:

Reason Type	Global Reason Codes	Team Specific Reason Codes
Not-ready Reason Codes	Packaged CCE 2000 Agents Deployment - 100 Packaged CCE 4000 Agents Deployment - 1000 Packaged CCE 12000 Agents Deployment - 3000	Packaged CCE 2000 Agents Deployment - 100 Packaged CCE 4000 Agents Deployment - 100 Packaged CCE 12000 Agents Deployment - 100
Sign-out Reason Codes	Packaged CCE 2000 Agents Deployment - 100 Packaged CCE 4000 Agents Deployment - 1000 Packaged CCE 12000 Agents Deployment - 3000	Packaged CCE 2000 Agents Deployment - 100 Packaged CCE 4000 Agents Deployment - 100 Packaged CCE 12000 Agents Deployment - 100
Wrap-up Reason Labels	Packaged CCE 2000 Agents Deployment - 1500 Packaged CCE 4000 Agents Deployment - 1500 Packaged CCE 12000 Agents Deployment - 1500	Packaged CCE 2000 Agents Deployment - 100 Packaged CCE 4000 Agents Deployment - 100 Packaged CCE 12000 Agents Deployment - 100

Following are the REST responses received during the execution of REST API to configure the Reason Codes:

- **Success** - Configuration changes persist in AW DB and synchronized with Finesse DB.

  Code: 200

  Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with Finesse DB.

  Code: 201

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with Finesse DB failed.)

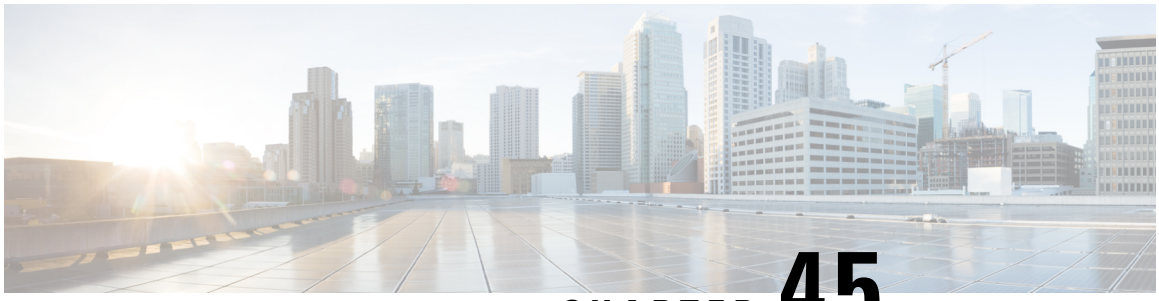
  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

```
<apiErrors>
<apiError>
<errorMessage>Configuration update failed for one or more devices.</errorMessage>
<errorType>PARTIAL_SUCCESS</errorType>
</apiError>
</apiErrors>
```

- **Failure**- The configuration updates to AW DB is failed.



CHAPTER 45

Role API

Roles specify the APIs that an administrator can use, and which menus and tools an administrator can see and use.

Use the Role API to list the roles currently defined in the database, or edit, delete, or create new custom roles.

Roles

Cisco Packaged CCE database has four built-in roles for administrators that set the access to specific features (APIs and tools). The built-in roles are hierarchical in that each role identified in the table below contains the access rights of the roles above it. You cannot alter the feature access for the built-in roles, but you can create additional roles to define customized sets of feature access.

This role	Allows full access to
AgentAdmin	Agent APIs and tools only.
ScriptAdmin	Agent, Call, and Scripting tools - including Bulk Jobs.
ConfigAdmin	All APIs and tools except administrator, departments, and roles.
SystemAdmin	All APIs and tools. Only this role provides access to the Administrator, Department, and Role features.

For more information on these roles, see the *Cisco Packaged Contact Center Enterprise Administration and Configuration Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html.

URL

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

Operations

- create: Creates one role, given the specified name, description, and featureList.
- delete: Deletes one role permanently from the database.
- get: Returns one role using the URL `https://<server>/unifiedconfig/config/role/<id>`.

- Auxiliary get method returns the accessList parameter containing a collection of all features that can be used to define a role. Use the URL `https://<server>/unifiedconfig/config/role/available_features`.

- list: Retrieves a list of roles.
- update: Gets the role item and updates the new configuration entered in the <accessList> fields.

Parameters

- refURL: The refURL of the role. See Shared Parameters, on page 8.
- name: The role name, such as ConfigAdmin. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- accessList: A collection of features that indicates which APIs and tools the administrators assigned to this role can access.
  - feature: Element of the access list that represents access to a feature (API / tool). Includes the following parameters:
    - name: Feature name. Maps to a valid API / tool.
    - category: Feature category, such as Agent, Call, System, and Access. This parameter is used only for get and list operations.
- administrators: A collection of administrator references (Administrator API, on page 15), including the user name, domain name, and refURL of the administrator. See References, on page 5.
- systemDefined: Read-only parameter. Indicates if this role is a system-defined role. Values are true/false.

Note

- When you enable the CampaignStatus or CampaignContact subfeature for a custom role then the Outbound Campaign API is provided with Update Only Access instead of Full Access. With Update Only Access, you cannot create and delete a campaign using Outbound Campaign API.
- When you enable the ManageAgentAttribute or ReSkillAgents subfeature for a custom role then the Agent API is provided with Update Only Access instead of Full Access. With Update Only Access, you cannot create and delete an agent using Agent API.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description

See Search, on page 10 and Sort, on page 11.

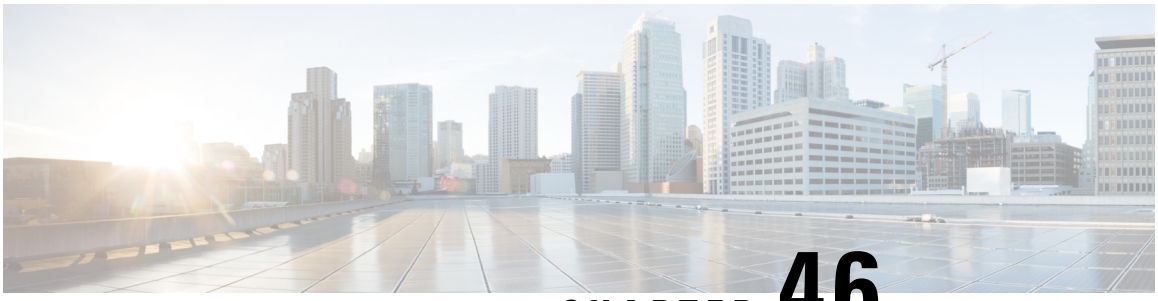
Advanced search parameters

You can perform a system defined role search on the Role API:

- **systemDefined:(both/only/none)**: Allows searching of roles with the specified value of the systemDefined parameter.
  - **q=systemDefined:only** Returns roles with the systemDefined parameter set to true.
  - **q=systemDefined:none** Returns roles with the systemDefined parameter set to false.
  - **q=systemDefined:both** Returns roles with the systemDefined parameter set to true or false.

Example Get Response

```
<role>
      <changeStamp>15</changeStamp>
      <refURL>/unifiedconfig/config/role/5003</refURL>
      <administrators>
        <administrator>
          <refURL>/unifiedconfig/config/administrator/5002</refURL>
          <domainName>BOSTON.COM</domainName>
          <userName>userAttribute</userName>
        </administrator>
      </administrators>
      <description>testAttribute</description>
      <accessList>
        <feature>
          <category>agent</category>
          <name>attribute</name>
        </feature>
        <feature>
          <category>agent</category>
          <name>reasoncode</name>
        </feature>
        <feature>
          <category>call</category>
          <name>bucketinterval</name>
        </feature>
      </accessList>
      <name>testAttribute</name>
      <systemDefined>false</systemDefined>
</role>
```



CHAPTER 46

Routing Type API

Use the Routing Type API to retrieve the list of routing types in the solution, including whether each routing type allows dialed number creation (Dialed Number API, on page 95). This API is read-only.

URL

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

Operations

- list: Returns routing type information for all data centers.
- get: Returns routing type information for a specific remote data center using the URL `https://<server>/unifiedconfig/config/routingtype?datacenter=<DatacenterName>`

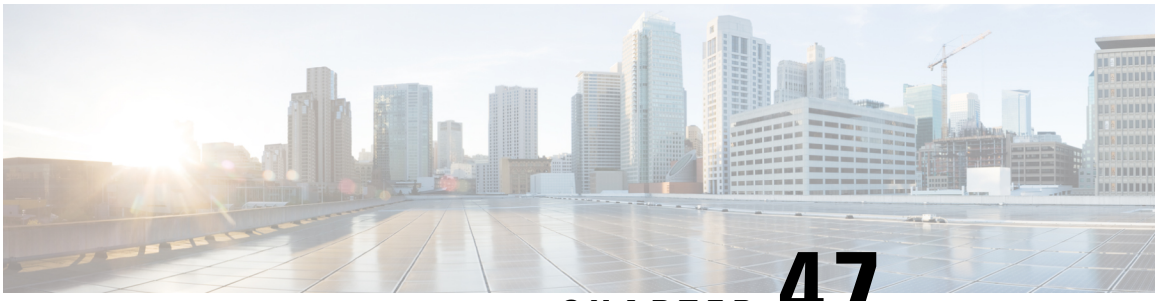
Parameters

- type: Indicates the value of this routing type (see Dialed Number API, on page 95).
- state: Indicates the state of the routing type:
  - UNKNOWN: Indicates that the system cannot determine the state of the routing type due to inventory or status errors.
  - UNUSED: Indicates that the routing type has not been used in Peripheral Gateway Setup.
  - MISCONFIGURED: Indicates that the routing type is used in Peripheral Gateway Setup, but either the Application Host Name associated with this routing type has not yet been added to the inventory, or there are status rule violations for the configuration.
  - OK: The routing type is configured correctly and dialed numbers can be created on this routing type (see Dialed Number API, on page 95).
- machineType: For routing types 4, 5, and 6, indicates the type of machine in the inventory that is used for this routing type. Potential machine types include EXTERNAL_SOCIAL_MINER , EXTERNAL_ECE, and EXTERNAL_THIRD_PARTY_MULTICHANNEL.
- datacenter: Indicates the datacenter to which the routing type is associated.

Example Get Response

```
<results>
    <routingTypes>
      <routingType>
```

```
        <type>1</type>
        <state>OK</state>
      </routingType>
      <routingType>
        <type>2</type>
        <state>OK</state>
      </routingType>
      <routingType>
        <type>3</type>
        <state>OK</state>
      </routingType>
      <routingType>
        <type>4</type>
        <machineType>EXTERNAL_ECE</machineType>
        <state>OK</state>
      </routingType>
      <routingType>
        <type>5</type>
        <state>MISCONFIGURED</state>
      </routingType>
      <routingType>
        <type>6</type>
        <state>UNUSED</state>
      </routingType>
    </routingTypes>
    <datacenterRoutingTypes>
      <datacenterRoutingType>
        <datacenter>
          <refURL>/unifiedconfig/config/datacenter/5000</refURL>
          <name>boston</name>
        </datacenter>
        <routingTypes>
          <routingType>
            <type>1</type>
            <state>OK</state>
          </routingType>
          ...
          ...
        </routingTypes>
      </datacenterRoutingType>
      ...
      ...
    </datacenterRoutingTypes>
</results>
```



CHAPTER 47

Routing Pattern API

The Routing Pattern API configures CVP routing pattern for the CVP server. The configured CVP routing pattern through API enables application user interface to configure routing pattern and the associated settings.

URL

`https://<server>:<serverport>/unifiedconfig/config/routingpattern`

Operations

- create: Creates a routing pattern.
- delete: Permanently deletes one routing pattern.
- get: Returns specific routing patterns from the database using the URL `https://<server>:<serverport>/unifiedconfig/config/ routingpattern/(id)`.
- list: Returns a list of routing patterns.
- update: Updates a routing pattern.

Parameters

- pattern: Required. Pattern for routing the call. The maximum length is 24 characters. It can contain alphanumeric characters, wildcard characters such as exclamation point (!) or asterisk (*), single digit matches such as the letter X or period (.). It can end with an optional greater than (>) wildcard character. This parameter cannot be updated.
- patternType: Required. Type of pattern. The value is 1 for VRU, 2 for Agents, and 3 for External.
- datacenter: A reference to the data center, including the refURL and name. This parameter cannot be updated.
- description: See Shared Parameters, on page 8.
- destination: Required. SIP Server Group/FQDN to which the pattern routes the call.
- sendToOriginator: Optional. Enables send calls to originator. The values are true or false. The calls are sent back to the originating Ingress gateway. This feature is not supported on Virtualized Voice Browser (VVB).

- rnaTimeout: Optional. Enables RNA Timeout for outbound calls. The value is an integer between 5 and 60.
- configParam: Optional. Miscellaneous futuristic field for future.
- changeStamp: See Shared Parameters, on page 8.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• pattern • description • destination	• pattern • datacenter.name • patternType • destination • description

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

The Routing Pattern API also supports advanced search parameters, such as patternType, datacenters, rnaTimeoutEnabled, and sendToOriginator.

- **patternType:** Finds all routing patterns that are associated to the specified pattern type - 1 for VRU, 2 for Agent, and 3 for External.
- **datacenters:(dc1|dc2|dc3):** Returns all routing patterns which belong to any of the specified data centers. You can specify up to three data centers. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all routing patterns in the core data center.
- **rnaTimeoutEnabled**: Finds all routing patterns for which RNA timeout is either configured or not. The possible values of the search parameter are as follows:
  - true - To search all routing patterns for which RNA timeout is configured.
  - false - To search all routing patterns for which RNA timeout is not configured.
- **sendToOriginator**: Finds all routing patterns for which Send to originator is either enabled or not. The possible values of the search parameter are as follows:
  - true - To search all routing patterns for which Send to originator is enabled.
  - false - To search all routing patterns for which Send to originator is not enabled.

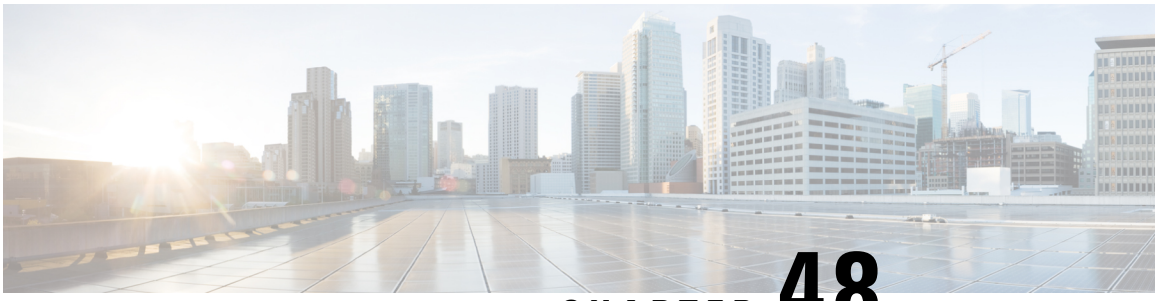
Example Create Request

```
<routingPattern>
    <pattern>xyz</pattern>
    <patternType>1</patternType>
    <destination>berlin.icm</destination>
    <datacenter>
```

```
        <name>Berlin</name>
        <refURL>/unifiedconfig/config/datacenter/10370</refURL>
    </datacenter>
    <description>Test</description>
    <rnaTimeout>30</rnaTimeout>
    <sendToOriginator>true</sendToOriginator>
    <configParam>abc</configParam>
</routingPattern>
```

Example Get Response

```
<routingPattern>
    <refURL>unifiedconfig/config/routingPattern/(id)</refURL>
    <changeStamp>0</changeStamp>
    <pattern>xyz</pattern>
    <patternType>1</patternType>
    <destination>berlin.icm</destination>
    <datacenter>
        <name>Berlin</name>
        <refURL>/unifiedconfig/config/datacenter/10370</refURL>
    </datacenter>
    <description>Test</description>
    <rnaTimeout>30</rnaTimeout>
    <sendToOriginator>true</sendToOriginator>
    <configParam>abc</configParam>
</routingPattern>
```



CHAPTER 48

Scan API

The Scan API forces an update of multiple types of system checks, including:

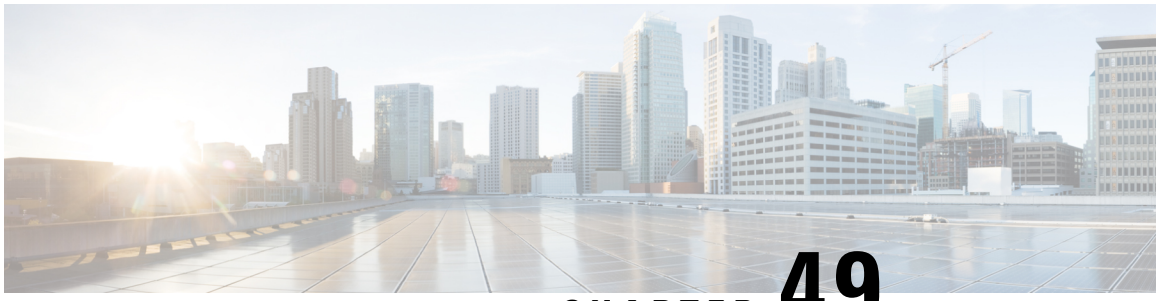
- Machine Inventory: Validate the VM Host configuration, VM layout, and required inventory configuration.
- Status: Validate the status rules. See Status API, on page 229.

URL

`https://<server>/unifiedconfig/config/status/scan`

HTTP method

PUT: Invoke the PUT method to force a scan to start immediately.



CHAPTER 49

Serviceability API

Use the Serviceability API to view information about the system, such as API statistics and version information.

URL

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

Operations

- get: Returns serviceability information.
  - **Query parameters:**
    - category: Use this query parameter to reduce the number of parameters returned. The values allowed match the names of the parameters. For example, `https://<server>/unifiedconfig/config/serviceability?category=systemValidationStatus&category=capacityInfo`.

Parameters

- currentTime: The time at which this web request was made.
- instanceName: The name of the active Unified CCE instance.
- version: Version information for Packaged CCE. Includes the following parameters:
  - 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.
- ucceVersion: Version information for Unified CCE. Includes the same parameters listed for the version parameter, above, as well as:

- versionString: Textual representation of the Unified CCE version.
- patchInfos: A collection of patch information, including majorVersion, minorVersion, maintenanceVersion, srVersion, and esVersion parameters.

- capacityInfo: A collection of capacityRules indicating if the capacity limits are valid. Each rule contains the following parameters:
  - name: The name of the capacity rule.
  - max: The maximum number of items allowed for the rule.
  - actual: The current number of items configured for the rule.

- systemValidationStatus: A collection of validationRules that show the potential errors regarding system configuration. For more information on the rules, see System Validation Rules, on page 201. Each rule contains the following parameters:
  - name: The name of the rule.
  - isValid: Indicates if the rule is passing. Values are true/false.
  - min: The minimum number of items required to match for this rule.
  - max: The maximum number of items required to match for this rule.
  - actual: The current number of items configured that match this rule.

Example Get 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>
    </patchInfos>
  </ucceVersion>
</Serviceability>
```

• System Validation Rules, on page 201

System Validation Rules

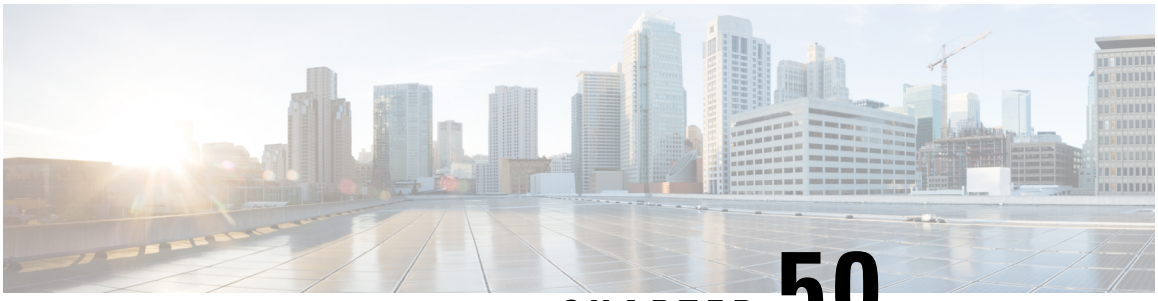
The system validation rules show the potential errors regarding system configuration.

Rule	Explanation
ECC_VARIABLES_CTI_SIZE	ECC Variables: Total bytes required for enabled variables in CTI Server must not exceed 2500 per call (2000 bytes for values and 500 bytes for variable names).
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.
PCCE_MR_PIM_COUNT	Peripheral: Exactly 4 Media Routing Peripherals must be configured on each Media Routing PG. This rule counts the quantity that are not compliant.
DESK_SETTING_WITH_RING_NO_ANSWER_SET_COUNT	Agent Desk Settings: Ring No Answer Timer must not be set.
ENT_SG_COUNT	Skill Groups: No Enterprise Skill Groups may be configured.
ENT_SG_MEMBER_COUNT	Skill Groups: No Enterprise Skill Group Members may be configured.
ENT_ROUTE_COUNT	Enterprise Routes: No Enterprise Routes may be configured.
ENT_ROUTE_MEMBER_COUNT	Enterprise Routes: No Enterprise Route Members may be configured.
PCCE_CUCM_PG_COUNT	Peripheral Gateway: There must be at least 1 Unified CM Peripheral Gateway configured but no more than 11.
VRU_PG_COUNT	Peripheral Gateway: There must be at least 1 VRU Peripheral Gateway configured but no more than 11.

Rule	Explanation
MR_PG_COUNT	Peripheral Gateway: There must be at least one Media Routing Peripheral Gateway configured but no more than 11.
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**.
UNSUPPORTED_PG_COUNT	Peripheral Gateway: Only PGs with CUCM, MediaRouting, and VRU client types are supported. This rule counts the quantity that are not compliant.
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 VRU Peripherals must be configured on a VRU PG and associated with the Type 10 Network VRU. This rule counts the quantity that are not compliant.
UCM_PIM_COUNT	Peripheral: Exactly 1 Unified CM Peripheral must be configured on each Unified CM PG in the Peripheral Explorer tool.
VRU_PIM_COUNT	Peripheral: Exactly 2 VRU Peripherals must be configured on each VRU PG.
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.

Rule	Explanation
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 in the PG Explorer tool must have exactly 1 label with a length of 10 digits. This rule counts the number of VRU PIMs that are not compliant.
CUCM_LABEL_COUNT	CUCM Routing Label: Each Unified CM Peripheral in the PG Explorer tool must have exactly 1 label with length of 10 digits. This rule counts the number of Unified CM Peripherals that are not compliant.
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.
PCCE_TYPE2_NETWORK_VRU_MAP_COUNT	Peripheral: Each MR PIM must be associated with the Type 2 Network VRU in the PG Explorer tool. This rule counts the quantity that are not compliant.
DIALED_NUMBER_EXTERNAL_VOICE_COUNT	Dialed Numbers: For each External Voice Dialed Numbers, there must be exactly 2 Dialed Number records for each Dialed Number String, with one for each VRU PIM.

Rule	Explanation
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.
NON_CUSTOM_APP_GATEWAY_COUNT	Application Gateway: No Application Gateways with application gateway type other than Custom Gateway may be configured.
DATABASE_LOOKUP_COUNT	Database Lookups: No Database Lookups may be configured.
SCRIPT_VERSIONS_ALLOWED	Max Script Versions: The maximum number of script versions allowed is 100 (minimum is 1).
OEM_CP_MATCHES_DB_COLLATION	The system locale must be compatible with the database collation. The valid pairs are: cp437 / iso_1 (US English) and cp850 / iso_1 (Other Latin derivatives). All other valid pairs must be identical strings: cp936 / cp936 (Chinese), cp866/cp866 (Russian), cp 932/cp932 (Japanese), and so on. For more information, see the Collation and Locale Settings for Localization section in the *Cisco Packaged Contact Center Enterprise Installation and Upgrade Guide* in the https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-installation-guides-list.html.
AGENT_TARGETING_RULE	Each Unified CM Peripheral must have exactly one agent targeting rule defined in CCE. This rule counts the quantity that are not compliant.



CHAPTER 50

Single Sign-On Global State API

Use the Single Sign-On (SSO) Global State API to view or update the global status of SSO.

To retrieve the overall status of setting the SSO state or the status for a single component, see the Single Sign-On Status API, on page 217.

URL

https://<server>/unifiedconfig/config/sso/globalstate

Operations

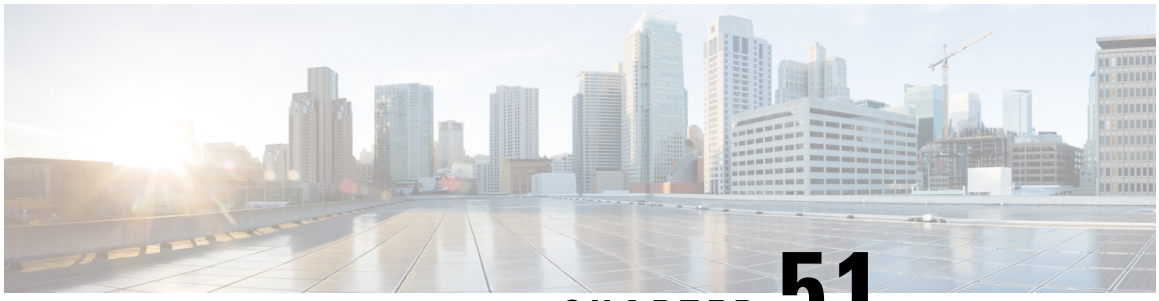
- get: Returns the current global state of SSO in the database.
- update: Updates the global state of SSO in the database.

Parameters

- refURL: The RefURL. See Shared Parameters, on page 8.
- changeStamp: See Shared Parameters, on page 8.
- permissionInfo: Information about permissions.
    - canUpdate: Whether ssoState can be updated. True or false.
    - role: The role of the user.
- state: Required for update. Valid values are NON_SSO (SSO is disabled for all users), SSO (SSO is enabled for all users), and HYBRID (mix of enabled and disabled).

Example Get Response

```
<ssoState>
    <refURL>/ssostate</refURL>
    <changeStamp>227</changeStamp>
    <permissionInfo>
        <canUpdate>true</canUpdate>
        <role>Administrator</role>
    </permissionInfo>
    <state>SSO</state>
</ssoState>
```



CHAPTER 51

Single Sign-On Registration API

Use the Single Sign-On (SSO) Registration API to register SSO-compatible components with the Cisco Identity Service. These components include AW, Finesse, and Unified Intelligence Center machines.

To retrieve the overall registration status or the status for a single component, see the Single Sign-On Status API, on page 217.

URL

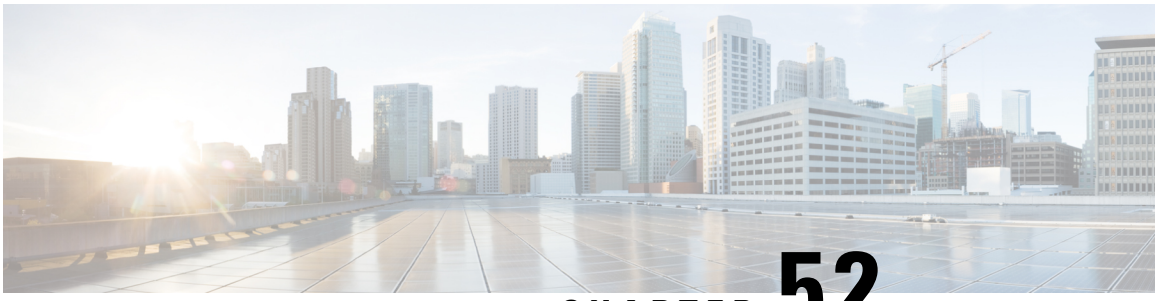
`https://<server>/unifiedconfig/config/sso/register`

Operations

- update: Registers all SSO-compatible components in the Machine Inventory with the Cisco Identity Service, using the URL `https://<server>/unifiedconfig/config/sso/register`. (See Machine Inventory API, on page 133

Parameters

None



CHAPTER 52

SIP Server Group API

You can add the SIP Server groups to perform SIP dynamic routing by Cisco Unified Customer Voice Portal (CVP).

A Server Group is identified by a Server Group domain name, also known as the Fully Qualified Domain Name (FQDN), and consists of one or more destination addresses (elements).

URL

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

Operations

- create: Creates a new SIP server group.
- delete: Permanently deletes one SIP server group.
- get: Returns specific SIP server groups from the database using the URL `https://<server>/unifiedconfig/config/sipservergroup/<id>`.
- list: Retrieves a list of SIP server groups.
- update: Updates one SIP server group.

Parameters

- name: Required. Name of the group (FQDN).
- description: See Shared Parameters, on page 8.
- type: Required. Type of group. This parameter accepts string. The value is:
  - VRU - For Virtualized Voice Browser (VVB), VXML Gateway, and Cisco Unified SIP Proxy (CUSP) devices.
  - Agent - For Cisco Unified Communications Manager (CUCM) and CUSP devices.
  - External - For Ingress Gateway and CUSP devices.
- datacenter: Site of the group. This is Null for core site. For remote sites, returns a reference to the data center, including the refURL and name. This parameter cannot be updated.
- defaultGroup: Whether this is the default group. This is read-only.

- elements: List the elements in this group. Includes the following parameters:
  - address: This is the valid hostname or IP address of the server group element.
  - port: Port number of the element in the server group.
  - secure port: The listening port for secure connection.
  - priority: Priority of the element in relation to the other elements in the server group. Specifies whether the server is a primary or backup server. Primary servers are specified as 1.
  - weight: Weight of the element in relation to the other elements in the server group. Specifies the frequency with which requests are sent to servers in that priority group.

Note You can add devices (elements) from different sites to a SIP Server group.

- noOfElements: Number of elements configured for a group. This parameter is used only for get operation.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name • description

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

The SIP Server Group API also supports advanced search parameters, such as sipServerType, datacenters, and groupElement.

- **sipServerType**: Finds all SIP Server groups that are associated to the specified group type - VRU, Agent, or External.
- **datacenters:(dc1|dc2|dc3):** Returns all routing patterns which belong to any of the specified data centers. You can specify up to three data centers. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all routing patterns in the core data center.
- **groupElement**: Finds the SIP Server group associated with the specified hostname or IP address of the element. The search is case-sensitive and does not support partial matches.

Example Get Response

```
<sipServerGroup>
            <refURL>/unifiedconfig/config/sipservergroup/5003</refURL>
            <changeStamp>0</changeStamp>
            <name>pcce.cisco.com</name>
            <type>Agent</type>
            <defaultGroup>false</defaultGroup>
 <datacenter>
                <refURL>/unifiedconfig/config/datacenter/9802</refURL>
```

```
            <name>Berlin</name>
        </datacenter>
        <elements>
            <element>
                <address>CCM-SUB-1B-131</address>
                <port>5060</port>
                <priority>10</priority>
                <refURL>/unifiedconfig/config/machineinventory/9792</refURL>
                <weight>10</weight>
            </element>
            <element>
                <address>CCM-SUB-1A-31</address>
                <port>5060</port>
                <priority>10</priority>
                <refURL>/unifiedconfig/config/machineinventory/9786</refURL>
                <weight>10</weight>
            </element>
        </elements>
        <noOfElements>2</noOfElements>
    </sipServerGroup>
```

Following are the REST responses received during execution of REST API to configure the SIP Server Group:

- **Success** - Configuration changes persist in AW DB and synchronized with respective devices.

  Code: 200

  Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with one or more devices.

  Code: 201

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with CVP failed.)

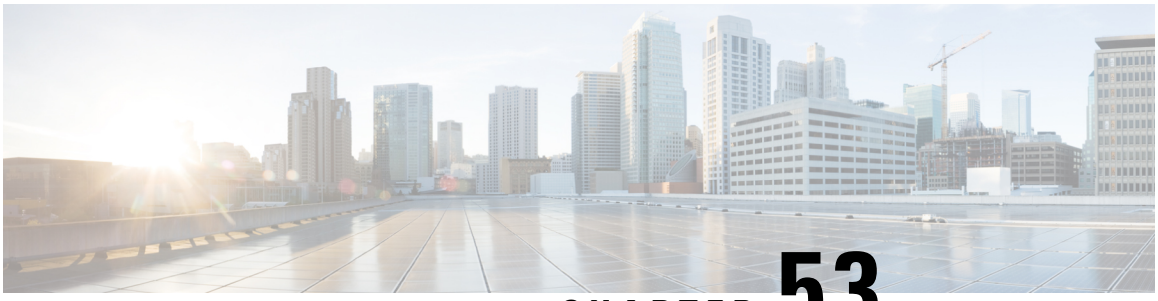
  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

```
<apiErrors>
<apiError>
<errorMessage>Configuration update failed for one or more devices.</errorMessage>
<errorType>PARTIAL_SUCCESS</errorType>
</apiError>
</apiErrors>
```

- **Failure**- The configuration updates to AW DB is failed.



CHAPTER 53

SIP Server Group Properties API

The SIP Server group properties provide a heartbeat mechanism that enables the originating endpoint to have the status of the destination endpoint before sending out the SIP request. Whether the destination is unreachable over the network, or is out of service, the originating SIP user agent knows the status through a heartbeat mechanism.

The Server group feature adds a heartbeat mechanism with endpoints for SIP. This feature allows faster failover on call control by eliminating delays due to failed endpoints.

Note The Up and Down Endpoint Heartbeat Interval is between any two heartbeats; however, it is not between heartbeats to the same endpoint. The SIP Server Group does not wake up at a specific interval and sends a heartbeat for all elements since this approach can result in CPU utilization issues. It also takes more resources to track heartbeats for many endpoints. For example, for 3 total elements across all SIP Server Groups, to proactively send a heartbeat to each element at 30000ms (30 seconds) intervals, you have to set the Endpoint Heartbeat Interval to 10000ms (10 seconds). It is less deterministic for reactive mode since elements that are currently down can fluctuate so the heartbeat interval fluctuates with it. To turn off pinging when the element is UP, set the UP interval to zero (reactive pinging). To turn off pinging when the element is down, set the DOWN interval to zero (proactive pinging). To ping when the element is either UP or DOWN, set both the intervals to greater than zero (adaptive pinging).

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

Operations

- get: Returns specific SIP server group properties from the database using the URL `https://<server>/unifiedconfig/config/sipservergroupproperties/<id>`.
- update: Updates one SIP server group properties.

Parameters

- serverGroupHeartbeats: Optional. Enable the heartbeat mechanism. The Heartbeat properties are editable only when this parameter value is true. The default value is false.
- serverGroupHBNumTries: Required. The number of failed heartbeats before marking the destination as unreachable.

  The default value is 3. Range is 1 to 5.

- serverGroupHBTimeout: Required. The amount of time, in milliseconds, before timing out the heartbeat.

  The default value is 800 milliseconds. Range is 100 to 3000.

- serverGroupUpInterval: Required. The ping interval for heart beating an endpoint (status) that is up.

  The default value is 5000 milliseconds. Range is 0 to 3600000

- serverGroupDownInterval: Required. The ping interval for heart beating an endpoint (status) that is down.

  The default value is 5000 milliseconds. Range is 0 to 3600000

- serverGroupHBLocalListenPort: Required. The heartbeat local socket listen port. Responses to heartbeats are sent to this port on CVP by endpoints.

  The default value is 5067. Range is 0 to 65535

- serverGroupHBMethod: Required. The heartbeat SIP method.

  The default value is OPTIONS. Supported values are OPTIONS and PING

- serverGroupHBTransportType: Required. During transportation, Server Group heartbeats are performed with a UDP or TCP socket connection. If unreachable or overloaded callbacks are invoked in the Server Group, that element is marked as being down for both UDP and TCP transports. When the element is up again, it is routable for both UDP and TCP.

  Note TLS transport is not supported.

- serverGroupOverloadedResponseCodes: Required. The response codes are used to mark an element as *overloaded* when received. If more than one code is present, it is presented as a comma-delimited list. An OPTIONS message is sent to an element and if it receives any of those response codes, then this element is marked as overloaded.

  The default value is 503,480,600. Maximum length of 128 characters, and accepts the number 0 to 9 and commas (,).

- optionsOverrideHost: Required. The contact header hostname to be used for a heartbeat request (SIP OPTIONS). The given value is added to the name of the contact header of a heartbeat message. Thus, a response to a heartbeat would contain gateway trunk utilization information.

  The default value is cvp.cisco.com. The maximum length of characters is 128.

Example Get Response

```
<CVP>
<sipServerGroupProperties>
<serverGroupHeartbeats>true</serverGroupHeartbeats>
<serverGroupHBNumTries>4</serverGroupHBNumTries>
<serverGroupHBTimeout>801</serverGroupHBTimeout>
<serverGroupUpInterval>5001</serverGroupUpInterval>
<serverGroupDownInterval>5000</serverGroupDownInterval>
<serverGroupHBLocalListenPort>5067</serverGroupHBLocalListenPort>
<serverGroupHBMethod>OPTIONS</serverGroupHBMethod>
<serverGroupHBTransportType>UDP</serverGroupHBTransportType>
<serverGroupOverloadedResponseCodes>200,300</serverGroupOverloadedResponseCodes>
<optionsOverrideHost>cvp.cisco.com</optionsOverrideHost>
```

```
</sipServerGroupProperties>
</CVP>
```

Following are the REST responses received during execution of REST API to configure the SIP Server group properties:

- **Success** - Configuration changes persist in AW DB and synchronized with respective devices.

  Code: 200

  Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with one or more devices.

  Code: 201

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with CVP Call Servers failed.)

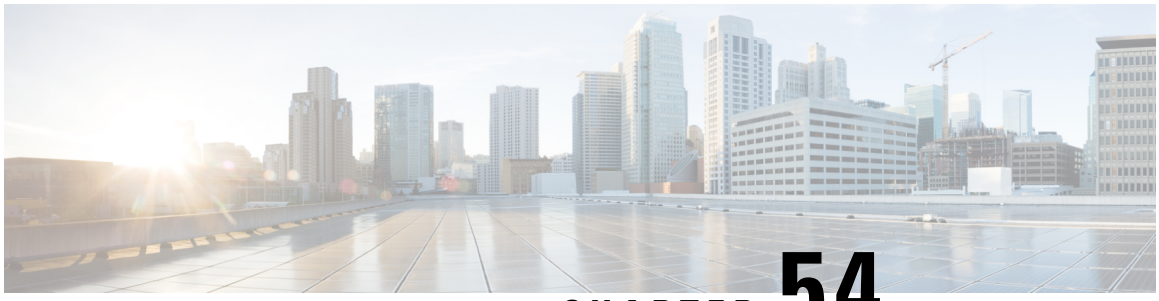
  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

  ```
  <apiErrors>
  <apiError>
  <errorMessage>Configuration update failed for one or more devices.</errorMessage>
  <errorType>PARTIAL_SUCCESS</errorType>
  </apiError>
  </apiErrors>
  ```

- **Failure**- The configuration updates to AW DB is failed.



CHAPTER 54

Single Sign-On Status API

Use the Single Sign-On (SSO) Registration API to get the current status of registering components with the Cisco Identity Service (IdS) and setting SSO state on the SSO-compatible components. These components include AW, Finesse, and Unified Intelligence Center machines.

URL

`https://<server>/unifiedconfig/config/sso/status`

Operations

- get component status: Returns the status of registering a specific component with the Cisco IdS and setting SSO state, using the URL `https://<server>/unifiedconfig/config/sso/status/<id>`. The <id> is the machine_id of the component.
- list: Retrieves a list with the overall and individual component statuses of registering SSO-compatible components with the Cisco Identity Service and setting SSO state, using the URL `https://<server>/unifiedconfig/config/sso/status`.

Parameters

- globalSsoState: The current SSO state as set in the AW database. The values are NON_SSO (SSO is disabled for all users), SSO (SSO is enabled for all users), and HYBRID (mix of enabled and disabled).
- registrationState: The overall status of registering components with the Cisco IdS. Values are the following:
  - SUCCEEDED: All of the components were successfully registered.
  - FAILED: Registration failed on one or more components. Error detail is set on each failed component.
  - PROCESSING: Registration started and is not complete.
  - NOT_STARTED: Registration has not started.
- modeState: The overall status of registering components with the Cisco IdS. Values are the following:
  - SUCCEEDED: The SSO state was successfully set on all of the components.
  - FAILED: The SSO state failed to be set on one or more components. Error detail is set on each failed component.

- PROCESSING: The SSO state change has started and is not complete.
- NOT_STARTED: The SSO state change has not started.

- idSConfigurationState: Whether the Cisco IdS has been configured and is in service.
  - STATE_NOT_CONFIGURED : The Cisco IdS is not been configured.
  - STATE_IN_SERVICE : The Cisco IdS is configured and is in service.
  - STATE_OUT_OF_SERVICE : The Cisco IdS is configured and is out of service.
  - STATE_PARTIAL_SERVICE: The Cisco IdS is configured and is partially in service.
  - STATE_UNREACHABLE: The Cisco IdS cannot be reached.
- hasIdsCredentials: Whether the Machine Inventory has the necessary IdS credentials to register components (see Machine Inventory API, on page 133. Values are true or false. The default is false.
- idsBaseUrl: The base URL for accessing the Identity Service.
- ssoComponentStatuses: A collection of registration and SSO state status information for all of the individual components. Returned on a list operation.
- ssoComponentStatus: Registration and SSO state status information for an individual component. Includes the following parameters:
  - registrationState: The status of registering the component with the IdS. See the values for this parameter above.
  - modeState: The status of setting the SSO state for the component. See the values for this parameter above.
  - refURL: The refURL for the component machine. See Shared Parameters, on page 8.
  - name: The name of the component machine.

Example Get Response

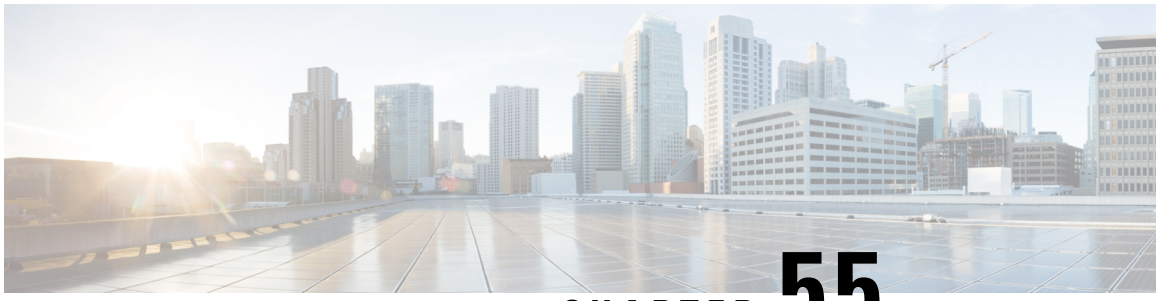
Example URL: `https://<server>/unifiedconfig/config/sso/status/21`

```
<ssoComponentStatus>
  <registrationState>FAILED</registrationState>
  <modeState>NOT_STARTED</modeState>
  <refURL>/unifiedconfig/config/machineinventory/21</refURL>
  <name>FINESSE-A.boston.com</name>
</ssoComponentStatus>
```

Example List Response

```
<ssoStatus>
  <globalSsoState>HYBRID</globalSsoState>
  <registrationState>FAILED</registrationState>
  <modeState>NOT_STARTED</modeState>
  <idSConfigurationState>STATE_IN_SERVICE</idSConfigurationState>
  <hasIdsCredentials>true</hasIdsCredentials>
  <idsBaseUrl>https://<server>:<serverport></idsBaseUrl>
  <ssoComponentStatuses>
```

```
    <ssoComponentStatus>
      <registrationState>FAILED</registrationState>
      <modeState>NOT_STARTED</modeState>
      <refURL>/unifiedconfig/config/machineinventory/21</refURL>
      <name>FINESSE-A.boston.com</name>
    </ssoComponentStatus>
    <ssoComponentStatus>
      <registrationState>FAILED</registrationState>
      <modeState>NOT_STARTED</modeState>
      <refURL>/unifiedconfig/config/machineinventory/22</refURL>
      <name>FINESSE-B.boston.com</name>
    </ssoComponentStatus>
    <ssoComponentStatus>
      <registrationState>FAILED</registrationState>
      <modeState>NOT_STARTED</modeState>
      <refURL>/unifiedconfig/config/machineinventory/23</refURL>
      <name>CUIC-A.boston.com</name>
    </ssoComponentStatus>
  </ssoComponentStatuses>
</ssoStatus>
```



CHAPTER 55

Skill Group API

A skill group is a collection of agents who share a common set of competencies that equip them to handle the same types of requests. Some examples of skill groups are a collection of agents who speak a specific language or who can assist callers with billing questions.

Use the Skill Group API to list the skill groups currently defined in the database, define new skill groups, and view, edit, or delete existing skill groups.

Note Access to this API is different for supervisors and administrators. For more information, see Access, on page 4.

URL

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

Operations

- create: Creates one skill group.

  Note A skill group can only be associated with agents that are on the same data center as that skill group.

- delete: Marks one skill group for deletion, but does not permanently delete it.
- get: Returns one skill group, using the URL `https://<server>/unifiedconfig/config/skillgroup/<id>`.
- list: Retrieves a list of skill groups.
  - **Query Parameters:**
    - selectedAgentCount: Use this query parameter to augment skill group information about multiple agents. The selectedAgentCount parameter shows the number of specified agents belonging to that skill group. For example, to find out how many of agents 5000, 5001, 5002, and 5003 belong to each of the skill groups in the list, add **selectedAgentCount=5000,5001,5002,5003**.

Note Using selectedAgentCount automatically sets the summary list query parameter to **true**.

- Summary list: See list, on page 3.

- update: Updates one skill group.

Note
- After a skill group has been created, the data center cannot be modified.
- A skill group can only be associated with agents that are on the same data center as that skill group.

Parameters

- refURL: The refURL of the skill group. See Shared Parameters, on page 8.
- name: The name of the skill group. See Shared Parameters, on page 8.
- department: A reference to the department (Department API, on page 89), including the name and refURL. See References, on page 5.
- changeStamp: See Shared Parameters, on page 8.
- description: See Shared Parameters, on page 8.
- mediaRoutingDomain: A reference to the media routing domain (Media Routing Domain API, on page 143) including the name and refURL. See References, on page 5.
  - Defaults to Cisco_Voice MRD if this parameter is not provided.
  - This reference cannot be updated.
- agents: A collection of agents assigned to the skill group (See Agent API, on page 17). References also include firstName, lastName, agentId, and agentTeam (which includes the team name and refURL). See References, on page 5.
  - canRemove: This parameter only appears for supervisors. It indicates whether or not 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.
- agentsAdded: A collection of agent references to be added to the skill group, including the refURL of each agent to be added. This parameter is update only, and cannot be used in conjunction with the agents parameter. This parameter can be used with the agentsRemoved parameter. See References, on page 5.
- agentsRemoved: A collection of agent references to be removed from the skill group, including the refURL of each agent to be removed. This parameter is update only, and cannot be used in conjunction with the agents parameter. This parameter can be used with the agentsAdded parameter. See References, on page 5.

- agentCount: Read-only parameter containing the number of agents having the skill.
- selectedAgentCount: Read-only field. Indicates the number of specified agents belonging to the skill group. Returned only when using the selectedAgentCount query parameter.
- bucketInterval: A reference to the bucket interval (Bucket Interval API, on page 47). Includes the name and refURL. See References, on page 5.
- serviceLevelThreshold: Maximum time in seconds that a caller should wait before being connected with an agent. Positive integers only, or blank.

  Blank means use the value from the specified mediaRoutingDomain.
- serviceLevelType: This value indicates how the system calculates the service level.
  - 1: Ignore Abandoned Calls (default).
  - 2: Abandoned Calls have Negative Impact.
  - 3: Abandoned Calls have Positive Impact.
- peripheralSet: This parameter is mandatory for Packaged CCE 4000 Agents or 12000 Agents deployment type. You must provide the reference to a peripheral set for which Agent PG is configured.

  The peripheralSet parameter is not available for Packaged CCE 2000 Agents deployment type.
- peripheralNumber: Read-only parameter. Automatically generated when using the create operation.
- datacenter: A reference to the data center, including the refURL and name.

  You must provide the reference to a data center that contains above peripheral set. For more information on data center for 4000 Agents or 12000 Agents deployment, see Inventory Import API, on page 123.

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 records are updated and deleted via the Skill Group API.

Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description • serviceLevelThreshold • serviceLevelType • peripheralNumber • datacenter.name • peripheralSet.name (Available for Packaged CCE 4000 Agents and 12000 Agents deployment types)

See Search, on page 10 and Sort, on page 11.

For more information on search restrictions, see Search, on page 10.

Advanced Search Parameters

- **datacenters: (dc1|dc2|dc3...)** which returns all the Skill Groups that belong to any of the specified data centers. Up to three data centers can be specified. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all machines in the core data center.
- **campaign:(none)** which returns all the Skill Groups that are not associated to any campaign.
- **peripheralSets: (ps1|ps2|ps3...)** returns the skill groups specific to the peripheral sets in 4000 Agents/12000 Agents deployment. The peripheral set names are fully matched (case-insensitive, no partial matches).

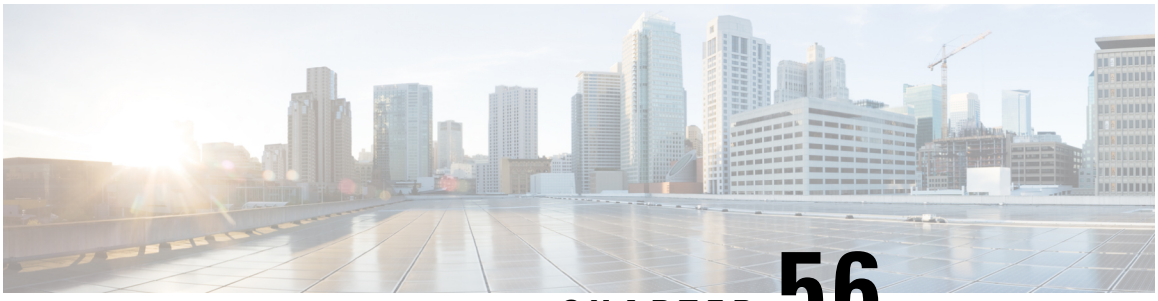
Example Get Response

```
<skillGroup>

<refURL>/unifiedconfig/config/skillgroup/(id)</refURL>
    <name>test</name>
    <description>test skill group</description>
    <changeStamp>0</changeStamp
    <datacenter>
        <name>Berlin</name>
        <refURL>unifiedconfig/config/datacenter/5000</refURL>
    </datacenter>
    <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>
            <userName>username</userName>
            <agentId>8007</agentId>
            <canRemove>true</canRemove>
        </agent>
        <agent>
            <refURL>/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>
    <agentCount>4</agentCount>
 </skillGroup>
```

Example Get Response for Packaged CCE 4000 Agents or 12000 Agents Deployment Type

```
<skillGroups>
<skillGroup xsi:type="skillGroup">

<refURL>/unifiedconfig/config/skillgroup/8485</refURL>
<changeStamp>0</changeStamp>
<agentCount>1</agentCount>
<name>SKG1</name>
<peripheralNumber>9194364</peripheralNumber>
<agents>
    <agent>...</agent>
</agents>
<PeripheralSet>

<refURL>/unifiedconfig/config/inventory/datacenter/bangalore/peripheralset/5001</refURL>
            <name>ps1</name>
 <PeripheralSet/>
 <datacenter>
        <refURL>/unifiedconfig/config/inventory/datacenter/10788</refURL>
        <name>bangalore</name>
 </datacenter>
<mediaRoutingDomain>...</mediaRoutingDomain>
</skillGroup>
</skillGroups>
```



CHAPTER 56

Stats API

Use the Stats API to get statistical information about your deployment, such as the number of logged in agents. This API is read-only.

URL

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

Operations

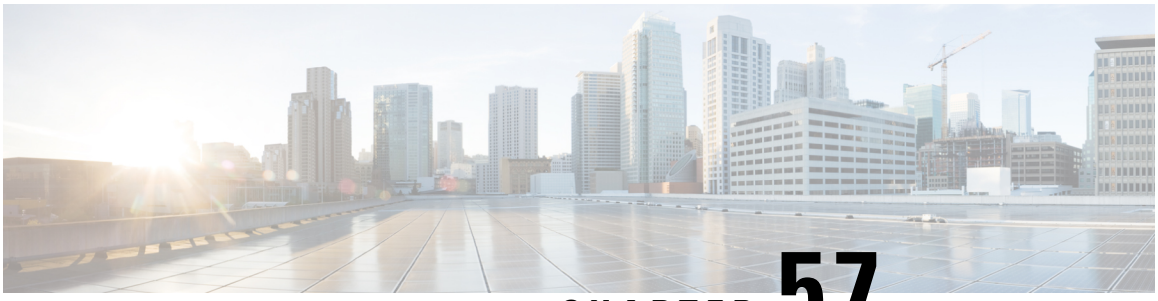
- get: Returns statistical information about your deployment.

Response Parameters

- numberOfAgentsLoggedIn: The number of agents logged in.

Example Get Response

```
<stats>
    <numberOfAgentsLoggedIn>10</numberOfAgentsLoggedIn>
</stats>
```



CHAPTER 57

Status API

The Status API retrieves information about the state of rules that collectively determine if the system is working correctly.

- Configuration Rules, on page 230
- Operation Rules, on page 237
- System Health Rules, on page 244
- VM Rules, on page 245

The system periodically checks the state of the rules. To force the status to be updated, refer to the Scan API, on page 197.

URL

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

Operations

- get: Returns the status rule results, using the URL `https://<server>/unifiedconfig/config/status`.

Response Parameters

- name: The name of the status. See the Configuration Rules, on page 230, Operation Rules, on page 237, System Health Rules, on page 244, and VM Rules, on page 245.
- category: The status category of Configuration, Operation, System Health, or VM.
- level: The severity of the condition. Values include: OK, INFO, WARNING, ERROR, and BLOCKED. The BLOCKED level indicates that the rule has not been processed yet or that the failure of another rule prevents this rule from running.
- machine: A collection of machines (Machine Inventory API, on page 133) including each machine's name, type, and refURL. See References, on page 5.

Example Get Response

```
<results>
   <statuses>
      <status>
```

```
            <name>CPU_USAGE</name>
            <category>SYSTEM_HEALTH</category>
            <level>OK</level>
            <detail xsi:type="metricDataDetail"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
               <actual>2</actual>
            </detail>
            <machines>
               <machine>
                  <refURL>/unifiedconfig/config/machineinventory/7507</refURL>
                  <machineType>CVP</machineType>
                  <name>CVP-A</name>
                  <side>sideA</side>
               </machine>
            </machines>
         </status>
         <status>
            <name>MEMORY_USAGE</name>
            <category>SYSTEM_HEALTH</category>
            <level>OK</level>
            <detail xsi:type="metricDataDetail"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
               <actual>17</actual>
            </detail>
            <machines>
               <machine>
                  <refURL>/unifiedconfig/config/machineinventory/7507</refURL>
                  <machineType>CVP</machineType>
                  <name>CVP-A</name>
                  <side>sideA</side>
               </machine>
            </machines>
         </status>
      </statuses>
</results>
```


- Configuration Rules, on page 230
- Operation Rules, on page 237
- System Health Rules, on page 244
- VM Rules, on page 245

Configuration Rules

These rules show the potential configuration errors and warnings for all of the machines, ESX hosts, and Gateways. Each section has a rule table that applies to all machines listed in that category, as well as a rule table for each type of machine in that category.

CCE Machines

These rules show the potential configuration errors and warnings for CCE machines (Unified CCE Rogger, Unified CCE PG, Unified CCE PG on Remote Data Center, Unified CCE AW-HDS-DDS, Unified CCE External HDS).

Table 1: Rules for All CCE Machines (Unified CCE Rogger, Unified CCE PG, Unified CCE PG on Remote Data Center, Unified CCE AW-HDS-DDS, Unified CCE External HDS)

Rule	Description
TRACE_LEVEL_NORMAL	The trace level must be set to normal to ensure performance.
CAN_RESOLVE_ADDRESS_TO_FQDN	The machine address must be resolvable to a fully qualified domain name (FQDN).

Table 2: Unified CCE Rogger Rules

Rule	Description
LOGGER_INSTALLED	The logger service must be installed.
LOGGER_AUTOMATIC	The logger service startup type must be set to automatic.
NO_EXTRA_SERVICES_INSTALLED_ROGGER	Only required services can be installed.
ROUTER_INSTALLED	The router service must be installed.
ROUTER_AUTOMATIC	The router service startup type must be set to automatic.
ROUTER_APPGW_MUST_BE_ENABLED	Application Gateway must be enabled on the router using Unified CCE Web Setup.

Table 3: Unified CCE PG Rules

Rule	Description
CTI_SVR_INSTALLED	The CTI Server service must be installed.
CTI_SVR_AUTOMATIC	The CTI server service startup type must be set to automatic.
DIALER_INSTALLED	If installed, the dialer service must be installed on both sides.
DIALER_AUTOMATIC	If installed, the dialer service startup type must be set to automatic.
DIALER_INSTALLED_OUTBOUND_ENABLED	The dialer services must be installed when outbound is enabled.
MR_PG_MR_PIM_CONFIGURED_FOR_MULTICHANNEL_MACHINES	A Media Routing Peripheral must be configured in Peripheral Gateway Setup for each multichannel machine in the solution inventory. Multichannel machine types include: • EXTERNAL_SOCIAL_MINER • EXTERNAL_ECE • EXTERNAL_THIRD_PARTY_MULTICHANNEL
MR_PG_APPLICATION_SERVER_HOSTNAME_NO_DUPLICATES	Application Server Host Name for each peripheral must be unique.

Rule	Description
MR_PG_MULTICHANNEL_APPLICATION_SERVER _HOSTNAME_BOTH_SIDES_MATCH	Application Server Host Name for Media Routing Peripheral Multichannel must be the same on both sides.
MR_PG_MULTICHANNEL2_APPLICATION_SERVER _HOSTNAME_BOTH_SIDES_MATCH	Application Server Host Name for Media Routing Peripheral Multichannel2 must be the same on both sides.
MR_PG_MULTICHANNEL3_APPLICATION_SERVER _HOSTNAME_BOTH_SIDES_MATCH	Application Server Host Name for Media Routing Peripheral Multichannel3 must be the same on both sides.
MR_PG_MULTICHANNEL_APPLICATION_SERVERS _HOSTNAME_FOUND_IN_INVENTORY	Application Server Host Names used in Peripheral Gateway Setup for all multichannel peripherals must be created as external machines in the solution inventory. The machine type must be one of the following: • EXTERNAL_SOCIAL_MINER) • EXTERNAL_ECE • EXTERNAL_THIRD_PARTY_MULTICHANNEL
MR_PG_MR_PIM_COUNT	The number of media routing PG MR_PIM processes (mr_pim.exe) on Side A and Side B must match. Valid if 0 to 4 MR_PIMS are enabled.
MR_PG_INSTALLED	The Media Routing PG service must be installed.
MR_PG_AUTOMATIC	The Media Routing PG service must be set to automatic.
NO_EXTRA_SERVICES_INSTALLED_PG	Only required and optional services can be installed (extra services such as an additional PG or CTI Server are not permitted).
UCM_PG_APPUSER_BOTH_SIDES_MATCH	The Communications Manager PIM application user configured on Side A must be identical to the user configured on Side B.
UCM_PG_APPUSER_FOUND_ON_CUCM	The Communications Manager PIM application user must be configured on the Communications Manager as an application user.
UCM_PG_JTAPI_MATCHES_CM_SUB	The Communications Manager PIM service address configured in PG Setup must match the Communications Manager Subscriber address on the same side.
UCM_PG_JTAPI_CLIENT_VERSION _MATCH_UCM	The JTAPI Client version installed must match the JTAPI Client version available on Unified Call Manager.
UCM_PG_INSTALLED	The UCM PG service must be installed.
UCM_PG_AUTOMATIC	The UCM PG service startup type must be set to automatic.
VRU_PG_INSTALLED	The VRU PG service must be installed.
VRU_PG_AUTOMATIC	The VRU PG service startup type must be set to automatic.

Table 4: Unified CCE PG on Remote Data Center Rules

Rule	Description
When you configure Agent PG	
CTI_SVR_INSTALLED	The CTI Server service must be installed.
CTI_SVR_AUTOMATIC	The CTI server service startup type must be set to automatic.
DIALER_INSTALLED	If installed, the dialer service must be installed on both sides.
DIALER_AUTOMATIC	If installed, the dialer service startup type must be set to automatic.
UCM_PG_INSTALLED	The UCM PG service must be installed.
UCM_PG_AUTOMATIC	The UCM PG service startup type must be set to automatic.
UCM_PG_JTAPI_MATCHES_CM_SUB	The Communications Manager PIM service address configured in PG Setup must match the Communications Manager Subscriber address on the same side.
When you configure MR PG	
MR_PG_INSTALLED	The Media Routing PG service must be installed.
MR_PG_AUTOMATIC	The Media Routing PG service must be set to automatic.
When you configure VRU PG	
VRU_PG_INSTALLED	The VRU PG service must be installed.
VRU_PG_AUTOMATIC	The VRU PG service startup type must be set to automatic.
When you configure any type of PG	
NO_EXTRA_SERVICES_INSTALLED_PG	Only required and optional services can be installed (extra services such as an additional PG or CTI Server are not permitted).

Table 5: Unified CCE AW-HDS-DDS Rules

Rule	Description
SSO_COMPONENT_STATUS_MATCHES_GLOBAL	The global SSO status must be the same as component SSO status.
SSO_VALID_IDS_REFERENCE	If single sign-on is enabled, this machine must be associated with a valid Cisco Identity Service.
DISTRIBUTOR_CONAPI_MUST_BE_DISABLED	Configuration Management Service (CMS) Node and Agent Re-skilling Web Tool must be disabled using Unified CCE Web Setup.
DISTRIBUTOR_INSTALLED	The distributor service must be installed.

Rule	Description
DISTRIBUTOR_AUTOMATIC	The distributor service startup type must be set to automatic.
NO_EXTRA_SERVICES_INSTALLED_AW	Only required services can be installed.
TASK_ROUTING_APP_PATHS_EXIST	Each Peripheral Gateway with a Communications Manager PIM must have an associated Task Routing Application Path.

Table 6: Unified CCE External HDS Rules

Rule	Description
SSO_COMPONENT_STATUS_MATCHES_GLOBAL	The global SSO status must be the same as component SSO status.
SSO_VALID_IDS_REFERENCE	If single sign-on is enabled, this machine must be associated with a valid Cisco Identity Service.
DISTRIBUTOR_CONAPI_MUST_BE_DISABLED	Configuration Management Service (CMS) Node and Agent Re-skilling Web Tool must be disabled using Unified CCE Web Setup.
TASK_ROUTING_APP_PATHS_EXIST	Each Peripheral Gateway with a Communications Manager PIM must have an associated Task Routing Application Path.

Gateways

These rules show the potential configuration errors and warnings for gateways.

Table 7: Gateway Rules

Rule	Description
GW_CODEC	The dial peers for each CVP call server on each gateway are configured with the supported codec: • voice-class codec #num • dial-peer voice #num voip (codec).The supported voice codecs are g711alaw, g711ulaw, g729r8, mp4a-latm and g722-64 for the above four dial peers to CVP call servers (if the dial peers are configured on the voice gateway). The supported video codec is h264. The dial peer is identified via "session target ipaddress xxxxx". The IP address must point to the IP address of CVP call server.
GW_SIP_PROTOCOL	The dial peers for each CVP call server on each gateway must be configured with the supported session protocol, sipv2.

CUIC-LD-IdS Machines

These rules show the potential configuration errors and warnings for CUIC-LD-IdS machines.

Table 8: Rules for all CUIC-LD-IdS Machines

Rule	Description
SSO_COMPONENT_STATUS_MATCHES_GLOBAL	The global SSO status must be the same as component SSO status.
SSO_VALID_IDS_REFERENCE	If single sign-on is enabled, this machine must be associated with a valid Cisco Identity Service.
CAN_RESOLVE_ADDRESS_TO_FQDN	The machine address must be resolvable to an FQDN.

Table 9: Rules for the CUIC-LD-IdS Publisher

Rule	Description
CUIC_REALTIME_DS_CORRECT_HOST	The realtime datasource must be configured with the correct hosts: either the Side A and B Unified CCE AW-HDS-DDS Servers, or, if used, the External HDS.
CUIC_HISTORICAL_DS_CORRECT_HOST	The historical datasource must be configured with the correct hosts: either the Side A and B Unified CCE AW-HDS-DDS Servers, or, if used, the External HDS.

Finesse Machines

These rules show the potential configuration errors and warnings for Finesse machines.

Table 10: Rules for All Finesse Machines

Rule	Description
SSO_COMPONENT_STATUS_MATCHES_GLOBAL	The global SSO status must be the same as the component SSO status.
SSO_VALID_IDS_REFERENCE	If single sign-on is enabled, this machine must be associated with a valid Cisco Identity Service.
CAN_RESOLVE_ADDRESS_TO_FQDN	The machine address must be resolvable to an FQDN.

Unified CVP Machines

These rules show the potential configuration errors and warnings for CVP machines.

Table 11: Rules for All Unified CVP Machines (Unified CVP, Unified CVP OPS, Unified CVP Reporting)

Rule	Description
TRACE_LEVEL_NORMAL	The trace level must be set to normal to ensure performance.
CAN_RESOLVE_ADDRESS_TO_FQDN	The machine address must be resolvable to an FQDN.

Table 12: Unified CVP Rules

Rule	Description
CVP_CUCM_SIP_SERVER_GROUPS_MATCH _CUCM_FQDN	The CVP SIP Server Group names that contain Communication Manager addresses must match the Communication Manager Cluster Fully Qualified Domain Name.
CVP_CUCM_SIP_SERVER_GROUPS_ONLY _CONTAIN_CUCM_HOSTS	The CVP SIP Server Groups that contain Communication Manager addresses cannot contain non-Communication Manager addresses (which include Communication Manager hosts that are not part of the inventory).
CVP_RING_TONE_DN_PATTERN	The Ring Tone Dialed Number configured on the CVP Call Server should match the pattern 91*.
CVP_ERROR_LABEL_DN_PATTERN	The Error Tone Dialed Number configured on the CVP Call Server should match the pattern 92*.

Unified CM Machines

These rules show the potential configuration errors and warnings for Unified CM machines.

Table 13: Rules for All Unified CM Machines (Unified CM Publisher, Unified CM Subscriber)

Rule	Description
TRACE_LEVEL_NORMAL	The trace level must be set to normal to ensure performance.
CAN_RESOLVE_ADDRESS_TO_FQDN	The machine address must be resolvable to an FQDN.

Table 14: Unified CM Publisher Rules

Rule	Description
CUCM_FQDN_DEFINED	The Communication Manager Cluster Fully Qualified Domain Name must be defined.

Rule	Description
CVP_SERVER_MUST_HAVE_A_CM_SIP_TRUNK	Each CVP Server must be referenced by at least one Communications Manager SIP Trunk destination.

ESX Hosts

These rules show the potential configuration errors and warnings for ESX hosts.

Table 15: Rules for All ESX Hosts

Rule	Description
VMHOST_ESXI_VERSION_MATCH	Side A and Side B VM Hosts must be on the same ESXi version.
VM_DATASTORE	Virtual machines must be deployed on the correct datastore.

SocialMiner

These rules show the potential configuration errors and warnings for SocialMiner machines.

Table 16: SocialMiner Rules

Rule	Description
SOCIAL_MINER_MR_ENABLED	Multichannel routing must be enabled on SocialMiner.
CAN_RESOLVE_ADDRESS_TO_FQDN	The machine address must be resolvable to an FQDN.

Operation Rules

These rules show the potential operation errors and warnings for Unified CCE, Unified CM, Unified CVP, Gateways, Unified Intelligence Center, Finesse, and Enterprise Chat and Email. Each section has a rule table that applies to all machines listed in that category, as well as a rule table for each type of machine in that category.

Unified CCE Machines

These rules show the potential operation errors and warnings for Unified CCE machines (Rogger, PG, Unified CCE PG on Remote Data Center, and AW-HDS-DDS).

Table 17: Unified CCE Rogger Rules

Rule	Description
LOGGER_CAMPAIGN_MGR_RUNNING	The logger campaign manager process (campaignmanager.exe) must be running.
LOGGER_CONFIG_LOGGER_RUNNING	The logger configuration logger process (configlogger.exe) must be running.

Rule	Description
LOGGER_CSFS_RUNNING	The logger customer support forwarding service process (csfs.exe) must be running.
LOGGER_HIST_LOGGER_RUNNING	The logger historical logger process (histlogger.exe) must be running.
LOGGER_BA_IMPORT_RUNNING	The logger import process (baimport.exe) must be running.
LOGGER_RECOVERY_RUNNING	The logger recovery process (recovery.exe) must be running.
LOGGER_REPLICATION_RUNNING	The logger replication process (replication.exe) must be running.
LOGGER_RUNNING	The logger service must be running.
SERVER_CREDENTIALS	The Diagnostic Framework credentials entered for the Principal AW must be valid.
SERVER_CONNECTION	The Cisco ICM Diagnostic Framework service on this machine must be reachable on the network.
ROUTER_CCAGENT_INSVC_ACTIVE_ENABLE_COUNT	The router central controller agent process (ccagent.exe) must be in service for both PGs.
ROUTER_CCAGENT_RUNNING	The router central controller agent process (ccagent.exe) must be running.
ROUTER_DBAGENT_RUNNING	The router database agent process (dbagent.exe) must be running.
ROUTER_LIVE_DATA_ACTIVE_IDLE	The router Live Data connection must be active on one side and idle on the other side.
ROUTER_MDSPROC_IN_SVC_PR_ENB_DSB	The router message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
ROUTER_MDSPROC_RUNNING	The router message delivery service process (mdsproc.exe) must be running.
ROUTER_ROUTER_RUNNING	The router process (router.exe) must be running.
ROUTER_RUNNING	The router service must be running.

Table 18: Unified CCE PG Rules

Rule	Description
CTI_SVR_CTI_SVR_ACTIVE_IDLE	The CTI server process (ctisvr.exe) must be active on one side and idle on the other side.

Rule	Description
CTI_SVR_CTI_SVR_RUNNING	The CTI server process (ctisvr.exe) must be running.
CTI_SVR_RUNNING	The CTI Server service must be running.
DIALER_RUNNING	If dialer is installed, then the dialer service must be running.
DIALER_BA_DIALER_SIP_ACTIVE_IDLE	The dialer process (badialer_sip.exe) must be active on one side and idle on the other side.
DIALER_BA_DIALER_SIP_RUNNING	The dialer process (badialer_sip.exe) must be running.
MR_PG_MR_PIM_ACTIVE_IDLE	Each MR_PIM process (mr_pim.exe) must be active on one side and idle on the other side.
MR_PG_MDSPROC_IN_SVC_PR_ENB_DSB	The media routing PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
MR_PG_MDSPROC_RUNNING	The media routing PG message delivery service process (mdsproc.exe) must be running.
MR_PG_PG_AGENT_ACTIVE_IDLE	The media routing PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.
MR_PG_PG_AGENT_RUNNING	The media routing PG PG agent process (pgagent.exe) must be running.
MR_PG_RUNNING	The media routing PG service must be running.
SERVER_CREDENTIALS	The Diagnostic Framework credentials entered for the Principal AW must be valid.
SERVER_CONNECTION	The Cisco ICM Diagnostic Framework service on this machine must be reachable on the network.
UCM_PG_JTAPI_ACTIVE_IDLE	The UCM PG jtapi process (jtapigw.exe) must be active on one side and idle on the other side.
UCM_PG_JTAPI_RUNNING	The UCM PG jtapi process (jtapigw.exe) must be running.
UCM_PG_LIVE_DATA_ACTIVE_IDLE	The UCM PG Live Data connection must be active on one side and idle on the other side.
UCM_PG_MDSPROC_IN_SVC_PR_ENB_DSB	The UCM PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
UCM_PG_MDSPROC_RUNNING	The UCM PG message delivery service process (mdsproc.exe) must be running.

Rule	Description
UCM_PG_PIM_COUNT	The number of UCM PG's PIM processes (eagtpim.exe) on Side A and Side B must match. Valid if 1 PIM is enabled.
UCM_PG_PG_AGENT_ACTIVE_IDLE	The UCM PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.
UCM_PG_PG_AGENT_RUNNING	The UCM PG PG agent process (pgagent.exe) must be running.
UCM_PG_RUNNING	The UCM PG service must be running.
VRU_PG_VRU_PIM_ACTIVE_IDLE	Each VRU PIM process (vrupim.exe) must be active on one side and idle on the other side.
VRU_PG_MDSPROC_IN_SVC_PR_ENB_DSB	The VRU PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
VRU_PG_MDSPROC_IN_SVC_PR_ENB_DSB	The VRU PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
VRU_PG_MDSPROC_RUNNING	The VRU PG message delivery service process (mdsproc.exe) must be running.
VRU_PG_PIM_COUNT	The number of VRU PG's PIM processes (vrupim.exe) on Side A and Side B must match. Valid if 0 to 2 VRU PIMs are enabled.
VRU_PG_PG_AGENT_ACTIVE_IDLE	The VRU PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.
VRU_PG_PG_AGENT_RUNNING	The VRU PG PG agent process (pgagent.exe) must be running.
VRU_PG_RUNNING	The VRU PG service must be running.

Table 19: Unified CCE PG on Remote Data Center Rules

Rule	Description
When you configure Agent PG	
CTI_SVR_RUNNING	The CTI Server service must be running.
CTI_SVR_CTI_SVR_RUNNING	The CTI server process (ctisvr.exe) must be running.
CTI_SVR_CTI_SVR_ACTIVE_IDLE	The CTI server process (ctisvr.exe) must be active on one side and idle on the other side.

Rule	Description
DIALER_RUNNING	If dialer is installed, then the dialer service must be running.
DIALER_BA_DIALER_SIP_RUNNING	If dialer is installed, the dialer process (badialer_sip.exe) must be running.
DIALER_BA_DIALER_SIP_ACTIVE_IDLE	If dialer is installed, the dialer process (badialer_sip.exe) must be active on one side and idle on the other side.
UCM_PG_RUNNING	The UCM PG service must be running.
UCM_PG_LIVE_DATA_ACTIVE_IDLE	The UCM PG Live Data connection must be active on one side and idle on the other side.
UCM_PG_JTAPI_RUNNING	The UCM PG jtapi process (jtapigw.exe) must be running.
UCM_PG_JTAPI_ACTIVE_IDLE	The UCM PG jtapi process (jtapigw.exe) must be active on one side and idle on the other side.
UCM_PG_MDSPROC_RUNNING	The UCM PG message delivery service process (mdsproc.exe) must be running.
UCM_PG_MDSPROC_IN_SVC_PR_ENB_DSB	The UCM PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
UCM_PG_PG_AGENT_RUNNING	The UCM PG PG agent process (pgagent.exe) must be running.
UCM_PG_PG_AGENT_ACTIVE_IDLE	The UCM PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.
UCM_PG_PIM_COUNT	The number of UCM PG's PIM processes (eagtpim.exe) on Side A and Side B must match. Valid if 1 PIM is enabled.
When you configure MR PG	
MR_PG_RUNNING	The media routing PG service must be running.
When you configure VRU PG	
VRU_PG_RUNNING	The VRU PG service must be running.
VRU_PG_PG_AGENT_RUNNING	The VRU PG PG agent process (pgagent.exe) must be running.
VRU_PG_PG_AGENT_ACTIVE_IDLE	The VRU PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.

Rule	Description
VRU_PG_MDSPROC_RUNNING	The VRU PG message delivery service process (mdsproc.exe) must be running.
VRU_PG_MDSPROC_IN_SVC_PR_ENB_DSB	The VRU PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
VRU_PG_PIM_COUNT	The number of VRU PG's PIM processes (vrupim.exe) on Side A and Side B must match. Valid if 0 to 2 VRU PIMs are enabled.
VRU_PG_VRU_PIM_ACTIVE_IDLE	Each VRU PIM process (vrupim.exe) must be active on one side and idle on the other side.
When you configure any type of PG	
SERVER_CREDENTIALS	The Diagnostic Framework credentials entered for the Principal AW must be valid.
SERVER_CONNECTION	The Cisco ICM Diagnostic Framework service on this machine must be reachable on the network.

Table 20: Unified CCE AW-HDS-DDS and External HDS Rules

Rule	Description
DISTRIBUTOR_CONFIG_LOGGER_RUNNING	The distributor configuration logger process (configlogger.exe) must be running.
DISTRIBUTOR_RT_CLIENT_RUNNING	The distributor real-time client process (rtclient.exe) must be running.
DISTRIBUTOR_RT_DIST_RUNNING	The distributor real-time distributor process (rtdist.exe) must be running.
DISTRIBUTOR_RUNNING **Note** This rule is not applicable for the external HDS	The distributor service must be running.
DISTRIBUTOR_UPDATE_AW_RUNNING	The distributor update process (updateaw.exe) must be running.
DEPLOYMENT_TASKS_PASSING	The deployment tasks must all be passing.
SERVER_CREDENTIALS	The Diagnostic Framework credentials entered for the Principal AW must be valid.
SERVER_CONNECTION	The Cisco ICM Diagnostic Framework service on this machine must be reachable on the network.

Unified CM Machines

These rules show the potential operation errors and warnings for Unified CM machines.

Table 21: Unified CM Publisher and Unified CM Subscriber Rules

Rule	Description
SERVER_CREDENTIALS	The AXL service credentials entered for the Publisher must be valid.
SERVER_CONNECTION	The AXL service on the Publisher must be reachable on the network.

Unified CVP Machines

These rules show the potential operation errors and warnings for CVP machines.

Table 22: Rules for All Unified CVP Machines (Unified CVP, Unified CVP OPS, and Unified CVP Reporting)

Rule	Description
SERVER_CREDENTIALS	The Cisco CVP WebServicesManager credentials entered for the CVP Ops Console Server must be valid.
SERVER_CONNECTION	The Cisco CVP WebServicesManager service on CVP Ops Console Server must be reachable on the network.

Gateways

These rules show the potential operation errors and warnings for gateways.

Table 23: Gateway Rules

Rule	Description
SERVER_CREDENTIALS	The service credentials entered for the gateway in Unified CVP Ops Console must be valid.
SERVER_CONNECTION	The service must be reachable on the network.

Unified Intelligence Center Machines

These rules show the potential operation errors and warnings for Unified Intelligence Center machines.

Table 24: Rules for Unified Intelligence Center Machines (Unified Intelligence Center Publisher and Unified Intelligence Center Subscriber)

Rule	Description
SERVER_CREDENTIALS	The Unified Intelligence Center Administration credentials entered for the Publisher must be valid.

Rule	Description
SERVER_CONNECTION	The SOAP service on the Publisher must be reachable on the network.

Finesse

These rules show the potential operation errors and warnings for Finesse.

Table 25: Finesse Rules

Rule	Description
FINESSE_SYSTEM_STATUS	Finesse must be in service.
SERVER_CREDENTIALS	The Finesse Administration credentials entered for the Primary Finesse machine must be valid.
SERVER_CONNECTION	The SOAP service on the Primary Finesse machine must be reachable on the network.
FINESSE_TOMCAT_SERVICE_STARTED	The Tomcat service must be started.

SocialMiner Machines

These rules show the potential operation errors and warnings for SocialMiner machines.

Table 26: SocialMiner Rules

Rule	Description
SERVER_CREDENTIALS	The service credentials must be valid.
SERVER_CONNECTION	The service must be reachable on the network.

System Health Rules

System health data is collected on a 20 second interval. The values represent the 95th percentile calculated over the preceding 10 minute period.

Usage Errors and Warnings

The following system health rules show the usage errors and warnings for all virtual machines. The DISK_USAGE rule applies to all disks in the system.

Rule	Description	Machine Type
CPU_USAGE	CPU usage must be below 80%.	All
MEMORY_USAGE	Memory usage must be below 80%.	All

Rule	Description	Machine Type
DISK_USAGE	Disk usage must be below 90%. If any disk value is over the threshold, then the highest error level is reported.	All
Principal_AW_Status	To identify if the Principal AW machine is up or down. This rule is applicable only for Packaged CCE 4000 Agents and 12000 Agents Deployments	CCE_AW

Datastore Metrics

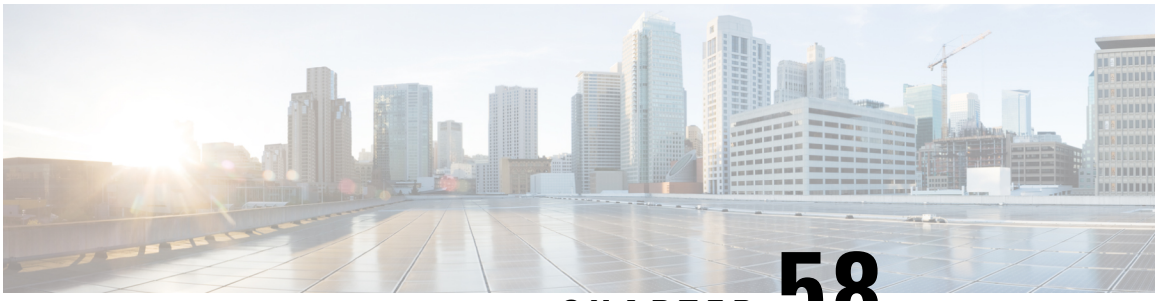
The following system health rules apply to all VM Hosts. Each rule output in the status detail identifies the datastore affected and the value for that particular metric.

Rule	Description
DATASTORE_DISK_COMMANDS_ABORTED_SUMMATION	Number of aborted commands per datastore.
DATASTORE_DISK_BUS_RESETS_SUMMATION	Number of bus resets per datastore.
DATASTORE_DISK_TOTAL_LATENCY_AVERAGE	Total latency in milliseconds per datastore.

VM Rules

These rules show the potential errors and warnings for virtual machines.

Rule	Description	Machine Type
VMWARE_TOOLS	VMware tools must be up-to-date.	VM Host
VMWARE_GUEST_OS	The operating system setting on each Virtual Machine (VM) must match the operating system installed on the Guest.	All



CHAPTER 58

Trace Level API

Use the Trace Level API to set the trace levels for the following components:

- Unified Contact Center Enterprise (CCE)
- Unified Customer Voice Portal (CVP)
- Unified Communications Manager (UCM)

URL

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

Operations

- list: Returns the trace level for each component type.
  - **Query parameters**
    - Summary list: See list, on page 3. Summary defaults to true on the Trace Level API.
- update: Sets the trace level for each component type.

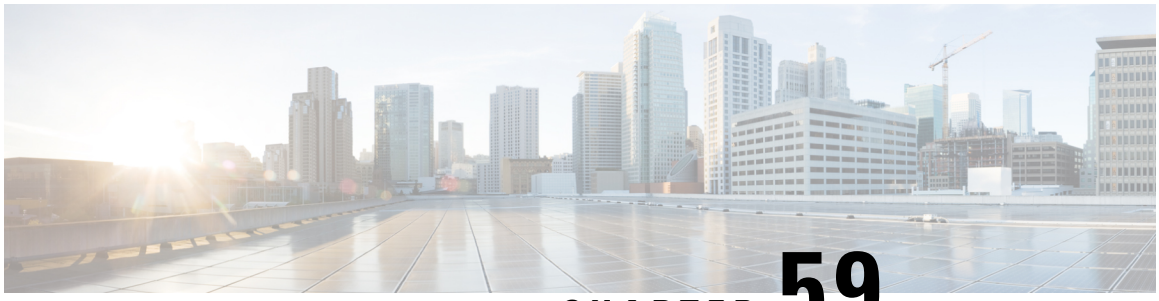
Parameters

- component: A list of components for which the trace level is set. Includes the following parameters:
  - type:
    - CCE
    - CVP
    - UCM
  - level:
    - NORMAL
    - DETAILED
    - CUSTOM: Not set by user. This level appears when one or more trace levels have been set by an outside program, as the component does not match the normal detailed definitions for the trace levels.

- traceMachines: A collection of trace information about each machine containing the refURL of the machine and a collection of process level trace values. Only available when the summary query parameter is false.

Example List Response

```
<traceLevels>
     <component>
        <type>CCE</type>
        <level>DETAILED</level>
     </component>
     <component>
        <type>CVP</type>
        <level>NORMAL</level>
     </component>
     <component>
        <type>UCM</type>
        <level>CUSTOM</level>
     </component>
</traceLevels>
```



CHAPTER 59

Transferable Files API

Use this API to upload files to AW. PCCE 12.0 and later only supports the upload of IVR application files.

Note This API only works on principal AW.

URL

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

Operations

- **get**: Returns one IVR file using the URL `https://<server>/unifiedconfig/config/transferablefiles`.
- **post**: Creates one IVR file.
- **delete**: Permanently deletes one IVR file.

Example Get Response

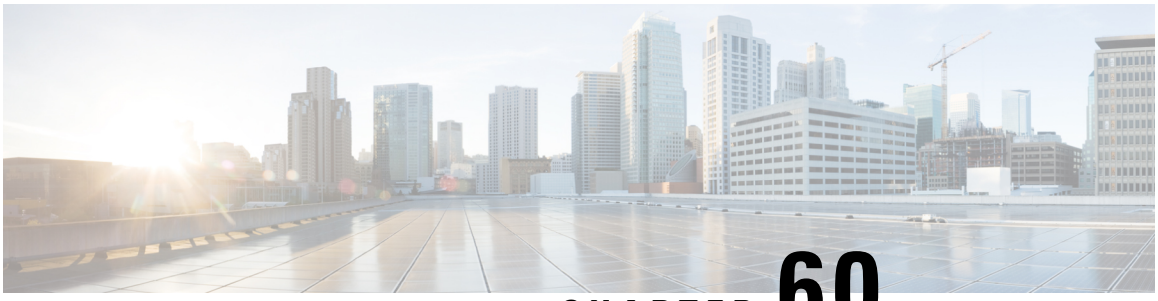
```
unifiedconfig/config/transferablefiles/ivrapplication
<results>
<ivrapplications>
<ivrapplication>
<     createDateTime>2019-04-10T08:22:58.371471Z</createDateTime>
<fileName>application2.zip</fileName>
<size>9511</size>
</ivrapplication>
<ivrapplication>
<createDateTime>2019-04-10T08:22:58.371471Z</createDateTime>
<fileName>application1.zip</fileName>
<size>9111</size>
</ivrapplication>
```

Example Post Response

```
unifiedconfig/config/transferablefiles/
The request has to have header
Content-Type: multipart/form-data;
```

Example Delete URL

```
unifiedconfig/config/transferablefiles/ivrapplication/<id>
```



CHAPTER 60

Uninitialize API

Use this API to roll back the changes made by the Initialization API to the system only if an initialization task is in the FAILED state.

- Start reversion process of initialization tasks which are in a state of SUCCEEDED or FAILED.
- Check the initialization status of the system using Initialization API.

URL

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

Operations

- list: Lists information about the system uninitialization status.
- update: Starts a system uninitialization.

Parameters

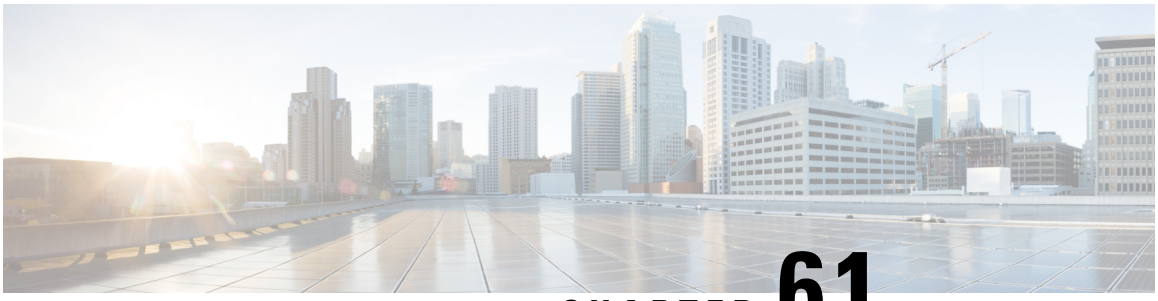
- initializationStatus: The name of the startup task.
- state: The state of the task. Values include PROCESSING, FAILED, FAILED_NEEDS_RETRY, or SUCCEEDED.

  Tasks in the FAILED_NEEDS_RETRY state do not require roll-back using the Uninitialize API.

Example GET Response

```
<results>
    <state>SUCCEEDED</state>
    <initializationStatuses>
        <initializationStatus><state>SUCCEEDED</state><name>SetProcessingStateTask</name>
        </initializationStatus>
        <initializationStatus><state>SUCCEEDED</state><name>SqlUserTask_sideA</name>
        </initializationStatus>
        <initializationStatus><state>SUCCEEDED</state><name>SqlUserTask_sideB</name>
        </initializationStatus>
        <initializationStatus><state>SUCCEEDED</state><name>DatabaseTask_sideA</name>
        </initializationStatus>
        <initializationStatus><state>SUCCEEDED</state><name>JtapiClientTask_sideA</name>
        </initializationStatus>
        <initializationStatus><state>SUCCEEDED</state><name>InstanceTask</name>
        </initializationStatus>
```

```
        <initializationStatus><state>SUCCEEDED</state><name>PGUserTask</name>
        </initializationStatus>
        <initializationStatus><state>SUCCEEDED</state><name>DatabaseTask_sideB</name>
        </initializationStatus>
        <initializationStatus><state>SUCCEEDED</state><name>JtapiClientTask_sideB</name>
        </initializationStatus>
    </initializationStatuses>
</results>
```



CHAPTER 61

Version API

The Version API is used to get the Unified Contact Center Enterprise version information from the system.

URL

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

Parameters

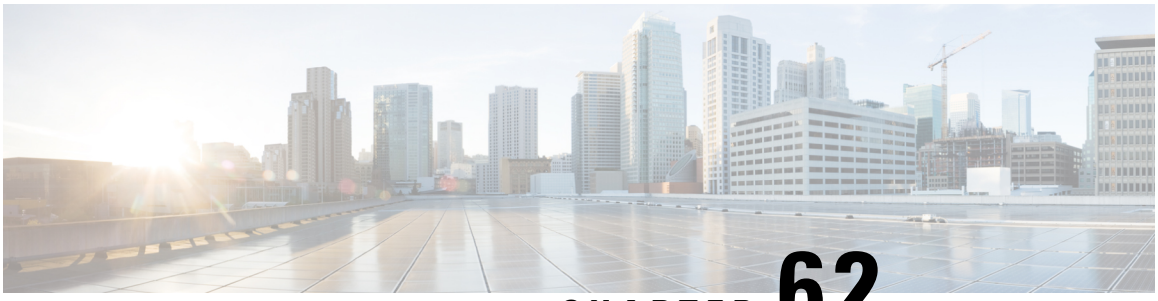
- ucceVersion: Information about the UCCE version installed on the system, including maintenance releases, engineering specials, and the schema version.

Operations

- get: Returns UCCE version information using the URL `https://<server>/unifiedconfig/config/version`.

Example Get Response

```
<versionInfo>
    <ucceVersion>
        <majorVersion>11</majorVersion>
        <minorVersion>0</minorVersion>
        <maintenanceVersion>1</maintenanceVersion>
        <srVersion>0</srVersion>
        <esVersion>0</esVersion>
        <buildVersion>3086</buildVersion>
        <versionString>11.0.1.0.0.3086</versionString>
        <schemaVersion>
            <major>181</major>
            <ccMinor>3</ccMinor>
            <awMinor>3</awMinor>
            <cceMinor>0</cceMinor>
        </schemaVersion>
    </ucceVersion>
</versionInfo>
```



CHAPTER 62

Virtualized Voice Browser Configuration API

The Cisco Virtualized Voice Browser (VVB) is an optional component that is configured as an external machine in the Packaged CCE System Inventory. Cisco VVB can be used as an alternative to the use of VXML gateways, which is to send the HTTP requests to CVP VXML server.

This API is used to configure the VVB parameters like Media and Security, Speech Servers, Applications properties, and fetch the configured data from AW DB.

URL

`https://<server>:<serverport>/unifiedconfig/config/deviceconfig/{site_name}/vvbserver`

Note For Main site, provide site_name as 'Core'.

Operations

- get: Gets the VVB device configuration data for the core or remote datacenter.
- update: Updates the VVB device configurations.

Parameters

- **Media Parameter**
  - **codec**: Required. Audio codecs are supported.
  - **MRCPVersion**: Required. Version of the MRCP protocol used to communicate between ASR/TTS server and Cisco VVB.

    Note ASR-TTS service is not supported using G729 codec; therefore, MRCP is not applicable for this codec.

  - **overrideSystemPrompt**: To override the system default prompt files. When enabled, the system plays the custom recorded prompt that is uploaded to the appropriate language directory under Prompt Management on Cisco VVB Admin UI.

Parameter	Default value	Supported values	Editable	VVB engine restart required after modification
codec	G711U	G711U G711A G729	Yes	Yes
MRCPVersion	MRCPv2	MRCPv1 MRCPv2	Yes	Yes
overrideSystemPrompt	false	true, false	Yes	Yes

- **Security Parameter**
  - **sipTLSEnabled**: Required. When enabled, this setting secures SIP signaling on the IVR leg.
  - **tlsVersion**: Required. Allows you to select one or more TLS versions of SIP. When you select a given TLS (SIP) version, Cisco VVB will support SIP TLS requests for this version and the higher supported versions.

    To enable this parameter, set the sipTLSEnabled parameter value to true.
  - **cipher**: Required. Defines the ciphers supported by Cisco VVB, with key size lesser than or equal to 1024 bits.

    To enable this parameter, set the sipTLSEnabled parameter value to true.
  - **srtpEnabled**: When SRTP (Secure Real-time Transport Protocol) is enabled, the IVR media is encrypted. The IVR leg is secured. SRTP uses Crypto-Suite AES_CM_128_HMAC_SHA1_32 for encrypting the media stream.

    To enable this parameter, set the sipTLSEnabled parameter value to true.
  - **allowMixedMode**: When Allow RTP (Mixed mode) is enabled, the VVB accepts both SRTP and RTP call flows. You can enable Allow RTP (Mixed mode) only when SRTP is enabled.

    To enable this parameter, set the sipTLSEnabled and srtpEnabled parameters value to true.

Parameter	Default value	Supported values	Editable	VVB engine restart required after modification
sipTLSEnabled	false	true, false	Yes	Yes
tlsVersion	TLSv1.2	TLSv1.0 TLSv1.1 TLSv1.2	Yes	Yes
cipher	TLS_RSA_WITH_AES_128_CBC_SHA - this cipher must be part of the cipher list as it is mandatory for TLSv1.2	Any other JDK supported ciphers	Add/Delete	Yes

Parameter	Default value	Supported values	Editable	VVB engine restart required after modification
srtpEnabled	false	true, false	Yes	Yes
allowMixedMode	false	true, false	Yes	Yes

- **Speech Server**
  - **asrServerName**: Required. Hostname or IP address of the ASR server.
  - **ttsServerName**: Required. Hostname or IP address of the TTS server.
- **Application & Triggers**

  By default, Packaged CCE will configure three applications - Comprehensive, Ringtone and Error. Below are the application specific parameters that you can edit.

  - **Comprehensive**

Parameter	Description
applicationName	Required. Application name - Comprehensive
sigDigits	The data type of the attribute is Integer. Enter the number of digits that are used as sigdigit. Range is 0 to 20. When Cisco VVB receives a call, the CVP comprehensive service is configured to strip the digits, so that when the IVR leg of the call is set up, the original label is used on the incoming VoiceXML request.
maximumSession	Required. Number of sessions to associate with the Comprehensive application. **Note** This number must not exceed the maximum number of ports supported for Cisco VVB profile. The data type of the attribute is integer. The default value is 600. Range is 1 to 600.
enableSecurity	Enable the option to encrypt the communication between Cisco VVB and VXML server. By default, this feature is disabled. The data type of the attribute is Boolean (Enable/Disable).

Parameter	Description
sipTriggers	SIP triggers required to invoke the application in response to incoming calls. **Note** Each time you invoke the API, the new list of SIP triggers overrides the existing triggers associated with the application. The data type of the attribute is string. Valid characters are alphanumeric (0-9)xXT and special characters *!>.+.

- **Ringtone**

Parameter	Description
applicationName	Required. Application name - Ringtone
maximumSession	Required. Number of sessions to associate with the Ringtone application. The data type of the attribute is integer. The default value is 600. Range is 1 to 600.
sipTriggers	SIP triggers required to invoke the application in response to incoming calls. **Note** Each time you invoke the API, the new list of SIP triggers overrides the existing triggers associated with the application. The data type of the attribute is string. Valid characters are alphanumeric (0-9)xXT and special characters *!>.+.

- **Error**

Parameter	Description
applicationName	Required. Application name - Error
maximumSession	Required. Number of sessions to associate with the Error application. The data type of the attribute is integer. The default value is 600. Range is 1 to 600.
customErrorPrompt	The custom error .wav file to play. The parameter is editable. **Note** Specify the custom prompt already uploaded into Cisco VVB, else default error prompt will be used.

Parameter	Description
sipTriggers	SIP triggers required to invoke the application in response to incoming calls. **Note** Each time you invoke the API, the new list of SIP triggers overrides the existing triggers associated with the application. The data type of the attribute is string. Valid characters are alphanumeric (0-9)xXT and special characters *!>.+.

Example Get Response

```
<VVB>
<media>
    <codec>G711U</codec>
    <MRCPVersion>MRCPv2</MRCPVersion>
    <overrideSystemPrompt>false</overrideSystemPrompt>
</media>
<security>
    <sipTLSEnabled>false</sipTLSEnabled>
    <tlsVersion>TLSv1.2</tlsVersion>
    <ciphers>
        <cipher>TLS_RSA_WITH_AES_128_CBC_SHA</cipher>
    </ciphers>
    <srtp>false</srtp>
    <allowMixedMode>false</allowMixedMode>
</security>
<Applications>
    <application>
        <applicationName>Comprehensive</applicationName>
        <sigDigits>0</sigDigits>
        <maximumSession>600</maximumSession>
        <enableSecurity>false</enableSecurity>
        <sipTriggers>
            <label>7777*</label>
            <label>8888*</label>
        </sipTriggers>
    </application>
    <application>
        <applicationName>Ringtone</applicationName>
        <maximumSession>600</maximumSession>
        <sipTriggers>
            <label>91919191*</label>
        </sipTriggers>
    </application>
    <application>
        <applicationName>Error</applicationName>
        <maximumSession>600</maximumSession>
        <sipTriggers>
            <label>92929292*</label>
        </sipTriggers>
    </application>
</Applications>
<asrServers>
    <asrServer>
        <asrServerName>10.10.10.10</asrServerName>
        <port>5060</port>
    </asrServer>
```

```
    <asrServer>
        <asrServerName>11.11.11.11</asrServerName>
        <port>5060</port>
    </asrServer>
</asrServers>
<ttsServers>
    <ttsServer>
        <ttsServerName>10.10.10.10</ttsServerName>
        <port>5060</port>
    </ttsServer>
    <ttsServer>
        <ttsServerName>11.11.11.11</ttsServerName>
        <port>5060</port>
    </ttsServer>
</ttsServers>
</VVB>
```

Following are the REST responses received during execution of REST API to configure the VVB devices for a given site:

- **Success** - Configuration changes persist in AW DB and synchronized with respective VVB devices.

  Code: 200

  Response: Successfully saved

- **Partial Success** - Configuration changes persist in AW DB, but failed to synchronize with one or more VVB devices.

  Code: 200

  Response: Configuration update failed for one or more devices. (This occurs when the AW DB is updated but Sync with VVB failed.)

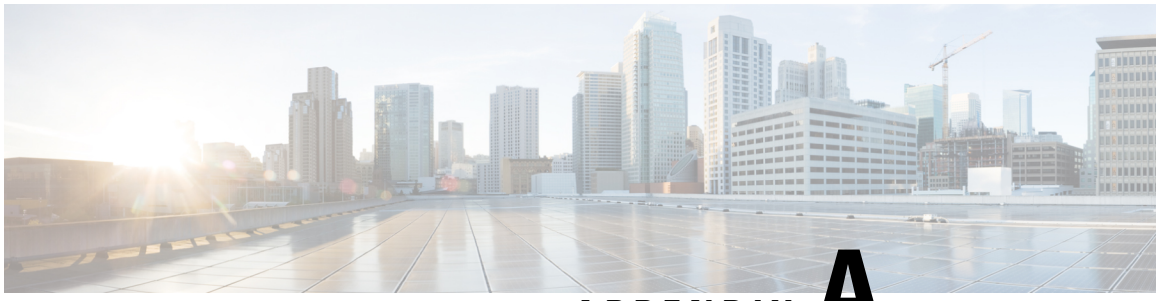
  Code: 503

  Response: The server is currently busy. Please try again later. (This occurs when data synchronization to a device is in progress.)

  Example API error:

  ```
  <apiErrors>
  <apiError>
  <errorMessage>Configuration update failed for one or more devices.</errorMessage>
  <errorType>PARTIAL_SUCCESS</errorType>
  </apiError>
  </apiErrors>
  ```

- **Failure**- The configuration updates to AW DB is failed.



APPENDIX A

Reference

- Third-party Integration UI Specification , on page 261

Third-party Integration UI Specification

Guidelines for Adding a Third-party User Interface

Follow these guidelines before adding a third-party user interface:

- You can only add third-party user interfaces that have a secure URL (https).
- A **Integrate as Gadget** check box is provided if the page needs to be a hosted by the Shindig framework.
- The URL of the user-interface must have the FQDN specified. URL with IPAddress is not allowed.
- The name of the user interface must be unique and its length is limited to 15 characters.
- Select any of the following system defined data to be used in the third-party user interface. These values are computed when the user-interface is rendered.
  - Deployment Type - Current deployment type (7 - PCCE 2K, 10 - PCCE 2K Lab, 17 -PCCE 4K, 18 - PCCE 12K)
  - Current User - The name and credentials of the logged on user. This data must be used as the authorization key to call the unifiedconfig API.
  - Current Role - The role of the logged on user. For example, administrator or supervisor.
  - API Base URL - API Base URL for unifiedconfig API. For example: `http://10.10.10.10/unifiedconfig/config/`
  - Locale - The current locale setting such as en_US.
- User-defined data is a key value pair.
  - Name can be used as a key.
  - Sensitive value can be masked by selecting the **Mask** check box. Once masked, the data cannot be unmasked or edited.

- The values of the key value pair will not be encoded in SPOG. This option only enables Unified CCE Administration to differentiate between sensitive and non-sensitive data and to report to the target application.
- If you want to use a coded value, you can use public key encryption to encode the value and set the value. Use private key in the target application server to decode it before it is used.
- **Unified CCE Administration** reports the sensitive fields list as a user preference (third-party user interface) or a query parameter.

Stylesheet Guidelines

Add the following style sheet into the third-party user interface.

```
/cceadmin/scripts/css/cd.component.css
```

Note Transferring the styles definition from the parent third-party user interface into the iframe through post message is not supported in Packaged CCE 12.0 or later.

Page Element	Styles	Class Names
Page Title	font-family: Cisco SansTT Light; font-size: 16px;	
Page Body	font-family: Cisco SansTT Regular; font-size: 12px;	
button (primary)	font-family: Cisco SansTT Regular; font-size: 14px;	bg-mint white bdr-transparent bdr-rad4 justify-content-ce align-item-center i-flex
button (secondary)	font-family: Cisco SansTT Regular; font-size:14px;	bg-gray white bdr-transparent bdr-rad4 justify-content-ce align-item-center i-flex
button (blue)	font-family: Cisco SansTT Regular; font-size:14px;	bg-gray white bdr-transparent bdr-rad4 justify-content-ce align-item-center i-flex
input[type=text] select	font-family: Cisco SansTT Regular; font-size:12px; min-width: 150px;	

Note For other UI elements styles, see the Business Hours pages in **Unified CCE Administration**.

Common FAQs

1. **How are system defined variables and user-defined variables passed to the third-party user interface?**
   - If **Integrate as Gadget** is checked, it will be constructed as **UserPrefs** using the Shindig container.
   - If **Integrate as Gadget** is not checked, the user interface is built and passed as a query string.
2. **Can I add a third-party user interface into a predefined menu such as the Agent menu?**

   Yes. The third-party user interface will be added as a tab. For example, Agent menu will be displayed in the first tab and the third-party user interface will be rendered as the second tab.

Troubleshooting

1. **Card does not appear on the Overview page to launch the third-party user interface.**
   - Card and menu appear after you login again.
   - Your role in Unified CCE Administration does not have access to the third-party user interface.
2. **The following error message appears in the screen:**

   ```
   Third-party user interface cannot be displayed. Please contact System
   Administrator
   ```

   The user defined data and system defined data are passed to the third-party user interface as query string. Length of the third-party user interface URL is limited to 2000 characters (URL includes address and query string). If the data has exceeded 2000 characters, minimize the length of the data passed as user defined and/or system defined data.
3. **Third-party user interface created as non-gadget (you did not select the "Integrated as a gadget" check box) is not rendering.**

   Webpages with http header x-frames set cannot be rendered inside another site's iframe.

Reference