



Cisco Unified Workforce Optimization

Quality Management API Programmer Guide Version 11.0

First Published: First Published: August 27, 2015

Last Updated: Last Updated: August 27, 2015

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

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

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

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

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

Quality Management API Programmer Guide

© 2015 Cisco Systems, Inc. All rights reserved.

Contents

Introduction	6
What's New in this Version	7
Server API	8
HTTPS Request Format	8
HTTP Request Header	9
Supported HTTP Methods	10
Authorization	10
Starting an Authorized Session	10
Ending an Authorized Session	11
Searching Contacts	11
Examples	18
Sample Contact Object	20
Exporting Recordings	21
Sample Request Body	22
Sample Response	22
Exporting Details	22
Sample Response	22
Downloading an Exported File	23
Deleting a Recording	23
Editing Metadata	23
Sample Request	24
Sample Response	24
Exporting a Single Contact Example	25
Recording Controls API	26
CAD and Finesse	26

Hot Desking	27
Recording Controls API Requirements	27
Agent Identifiers	27
Userdomain and Username	27
Peripheral_id and Sender_id	28
How to Find the Microsoft Windows Login Name	28
Rules for Recording Controls Commands	28
Recording Commands	32
Command Syntax	32
Command Functions	33
Effects of Issuing Recording Control Commands	47
Active and Last Call	48
Command Examples	50
Login Command Examples	50
Metadata Command Examples	50
Pause Command Examples	51
Resume Command Examples	51
Start Segment Command Examples	52
Using Commands with an Outbound Dialer	52
Integrating Recording Commands with CAD and Finesse	53
Configuring Recording Controls	53
Changing the Debugging Level	54
Pausing Audio and Screen Recording	55
Single Pause URL	55
Multiple Pause URLs	56
Assigning Pause URLs to Agents	57
Changing the Title	57
Configuring the IP Phone Service	57

Recording Controls IP Phone Service Considerations	58
Configuring Unified CM for IP Phone Service	58
Creating an IP Phone Service Definition	59
Assigning the IP Phone Service to Agents' Phones	59
Verifying that the Recording Controls IP Phone Service is Working ...	61
Recording Verification API	64
Recording Verification API Requirements	64
Recording Verification Commands	65
Command Syntax	65
Post-Call Survey API	68
Post-Call Survey API Requirements	68
CSV Format	69
Post-Call Survey Commands	70
Command Syntax	71
Contact Basic Search API	72
Contact Basic Search Commands	72
Command Syntax	72
Customer Relationship Management Integration	74
Best Practices	74

Introduction

This document describes the following application programming interfaces (API) for Cisco Unified Workforce Optimization Quality Management:

- Server API—allows users to search, export, edit, and delete Quality Management call data from uploaded contact recordings.
- Recording Controls API—allows agents to control which recorded calls are stored, the content of the recorded calls, and even the data associated with the calls
- Recording Verification API—allows users to locate calls and verify their recording status.
- Post-Call Survey API—allows users to import customer surveys into Recording and Quality Management
- Contact Basic Search API—allows users to search for details regarding an in progress or most recently completed call.

It also describes how to configure your customer relationship management (CRM) system to send URL commends to Quality Management.

What's New in this Version

Version	Changes
11.0(1)	<ul style="list-style-type: none">■ Combined the following documents into this document:<ul style="list-style-type: none">○ <i>Recording Controls API Programmer's Guide</i>○ <i>Recording Verification API Programmer's Guide</i>○ <i>Recording Services API Programmer's Guide</i>■ Added support for the Post-Call Survey API.■ Added support for the Contact Basic Search API■ Added HTTP Request Header■ Add information on Customer Relationship Management integration

Server API

The Server API allows users to search, export, edit, and delete Quality Management call data from uploaded contact recordings. The Server API is a REST-like API. The Server API uses Secure Sockets Layer (SSL), so uses HTTPS to issue requests. You format the request and response bodies using JavaScript Object Notation (JSON). For more information on JSON, and to access the JSON libraries, go to the following website:

<http://json.org/>

Note: Cisco only supports resources mentioned in this document. When you issue requests, you might see references to other resources. Undocumented resources are not supported and subject to change without warning.

HTTPS Request Format

The HTTPS request uses the following format:

```
https://<Server>/api/rest/recording/<resource>?<search query parameters>
```

Where <Server> is the IP address of the Quality Management server, <resource> is the type of request, and <search query parameters> is the search criteria used to filter results.

All requests require a date range or a specific contact ID.

List queries typically use search criteria to filter results and return a JSON array. The array contains a list of resources that contain references (\$ref) to resources that matched the specified search criteria. You can then choose to use one of the supported HTTP methods (GET, POST, PUT, and DELETE) to modify the references. Resources use the following format:

```
/api/rest/recording/<resource>/{id}
```

Where <resource> is the name of the resource and {id} is the unique identifier for that resource.

You can combine query parameters in a search and search for metadata keys. You can also search for metadata with a specific value.

HTTP Request Header

By default, the Server API results are limited, or paged, at 100 when you send a query.

The Range request header in an HTTP request limits the size of the response. This prevents large amounts of data from being returned in the response that the client may not be able to handle or expect. It is used by Unified Workforce Optimization to support paging in the user interface (UI). The UI requests a range of contacts and the response contains a Content-Range header that specifies the range that was returned and how many contacts total are in the request in order to calculate the number of pages. Unified Workforce Optimization's REST API for contacts adds a Range of 0-99 if no Range header is provided in the request. Without a default Range, it would be possible to return the entire contents of the contact database, possibly impacting client stability and server performance.

The format for the Range header value is:

```
items=<startIndex>-<endIndex>
```

The items are zero based so the first item is at index 0. So to return the first 20 contacts (items) from the request the request would be:

```
Range items=0-19
```

The response would contain a Content-Range header that contains the range that was actually returned and the total number of contacts available (in this case, 129 contacts). If the response contains the whole content, no Content-Range header is provided.

```
Content-Range items 0-19/129
```

To get all the items from the request (and override the default 0-99 range) you'll need to provide a sufficiently large endIndex to get the whole contents.

Maximum value for a range index is a signed 4 byte int (2147483647).

Range items=0-2147483647

Supported HTTP Methods

The Server API supports the following HTTP methods:

- GET—searches and retrieves saved information.
- POST—creates new resource. Use POST to create a new export or new login session.
- PUT—changes the existing data. Use PUT to modify metadata.
- DELETE—removes a resource.

Example: A recording.

Authorization

All Server API operations require an authorized session. The Server API requires the user ID and password of a user who is configured and licensed in Quality Management Administrator for authentication. The information in a response to a request only provides the information the user is allowed to see based on the roles and privileges assigned to the user in Quality Management Administrator.

Example: A supervisor only sees the contacts associated with teams assigned to the supervisor.

Starting an Authorized Session

To start an authorized session, send a POST request to `/api/rest/authorize`.

- If Quality Management is not configured for Active Directory, use the following format for the request body:

```
[{ "id": "recording", "userId": "<user name>",
  "password": "<password>" }]
```

Where <user name> and <password> are the user's Quality Management username and password.

- If Quality Management is configured for Active Directory, use the following format for the request body:

```
[{ "id": "recording", "userId": "<user name>",
  "password": "<password>", "data": {"domain": "<domain>" } ]]
```

Where <user name> and <password> are the user's Active Directory username and password, and <domain> is the user's Active Directory domain.

Ending an Authorized Session

To end an authorized session, send a DELETE request to `/api/rest/authorize`.

Searching Contacts

Use the GET method and the following resource URLs to search for contacts.

- `/api/rest/recording/contact/{id}`

Where {id} is the identifier for an existing contact. This resource URL locates a specific contact ID. If the {id} does not exist, the response returns an error code instead of an empty list.

- `/api/rest/recording/contact`

The following table describes the query parameters you can use when exporting contacts.

Query parameters

Parameter	Description
agent	A person's ID. The accepted value is integer ID. This parameter is optional.

Parameter	Description
ani	<p>The automatic number identification (ANI) for a call. In other words, ANI identifies the number of the calling party. The accepted value is string with asterisk (*) or question mark (?) wildcards.</p> <p>This parameter can appear zero or more times in a single query.</p> <div data-bbox="591 474 1383 575" style="border: 1px solid #ccc; border-radius: 10px; background-color: #e6f2ff; padding: 10px; margin: 10px 0;"> <p>Example: <code>contact?ani=1234&ani=4567</code></p> </div> <p>When you provide multiple values for a parameter, the query combines these values with OR (that is, the previous example searches for contacts with either <code>ani=1234</code> OR <code>ani=4567</code>). This parameter is optional.</p>
assocCallId	<p>The Associated Call ID that ties together contacts based on a customer experience. This parameter is optional.</p>
beginTime	<p>Return only records that start on or after the specified date.</p> <p>When specifying a date, choose one of the following date formats:</p> <ul style="list-style-type: none"> ■ YYYY-MM-DD HH:MM:SS ■ YYYY-MM-DD <p>This parameter is optional.</p>
canEvaluate	<p>Whether the person can evaluate this contact. The accepted value is true. True returns the contact, if the Quality Management user can evaluate this contact when logged into Unified Workforce Optimization.</p> <p>Accepted Values: “true” if this contact should be returned if the contact can be evaluate by the logged in person.</p> <p>This parameter is optional. If you do not include this parameter, the query does not filter the contacts.</p>

Parameter	Description
dnis	<p>The dialed number identification server (DNIS) for the call. In other words the called number. The accepted value is string with asterisk (*) or question mark (?) wildcards</p> <p>This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these values with OR. This parameter is optional.</p>
endTime	<p>Return only records that start before the specified date. When specifying a date, choose one of the following date formats:</p> <ul style="list-style-type: none"> ■ YYYY-MM-DD HH:MM:SS ■ YYYY-MM-DD <p>This parameter is optional.</p>
exclude	<p>Returns a Uniform Resource Identifier (URI) for the specified value. Click the URI to see the data associated with the specified value. The accepted values are:</p> <ul style="list-style-type: none"> ■ event—Returns a URI that points the event data. ■ metadata—Returns a URI that points to the metadata. ■ metadata.key1—Returns a URI that points to the key1 attribute within the metadata object. <p>This parameter is optional.</p>
expand	<p>Returns all data associated with the specified value instead of a URI. The accepted values are:</p> <ul style="list-style-type: none"> ■ event—Expands all events to include all event data, not just the URI. ■ metadata—Expands all metadata, not just the URI. ■ metadata.key1—Only expands the key1 attribute within the metadata object. <p>This parameter is optional.</p>

Parameter	Description
firstName	The agent's first name. The accepted values is string with any number of asterisk (*) or question mark (?) wildcards. This parameter is optional.
group	A group's ID. The accepted value is integer ID. This parameter is optional.
hasRecording	<p>Return only contacts associated with a recording. The accepted values are:</p> <ul style="list-style-type: none">■ true■ false <p>This parameter is optional. If you do not include this parameter, the query does not filter contacts by recordings.</p>
hr	<p>Whether the contact evaluation has been marked for human resources (hr). The accepted Boolean values are:</p> <ul style="list-style-type: none">■ true■ false■ 1 (true)■ 0 (false) <p>This parameter is optional.</p>
inProgress	<p>Whether the contact evaluation is in progress. The accepted Boolean values are:</p> <ul style="list-style-type: none">■ true■ false■ 1 (true)■ 0 (false) <p>This parameter is optional.</p>

Parameter	Description
lastName	<p>The agent's last name. The accepted values is string with any number of asterisk (*) or question mark (?) wildcards. This parameter is optional.</p> <p>This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these values with OR. This parameter is optional.</p>
line	<p>The extension for the call (from the perspective of the agent who is recording the call). The accepted value is string with asterisk (*) or question mark (?) wildcards.</p> <p>This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these values with OR. This parameter is optional.</p>
metadata	<p>The metadata field. The accepted values are:</p> <ul style="list-style-type: none"> ■ <key> ■ <key>~<value> <p>Where <key> is the name of the metadata field configured in Monitoring and Recording Administrator and must match the configured metadata name exactly. The <value> is optional and can include a string with asterisk (*) or question mark (?) wildcards. This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these values with OR. If you do not specify a value, the query returns all contacts that have metadata for the specified key.</p> <p>This parameter is optional.</p>

Parameter	Description
needsApproval	<p>Whether the contact evaluation needs approval. The accepted Boolean values are:</p> <ul style="list-style-type: none">■ true■ false■ 1 (true)■ 0 (false) <p>This parameter is optional.</p>
number	<p>Any number used in the contact (ANI, DNIS, or Line). The accepted value is string with asterisk (*) or question mark (?) wildcards.</p> <p>This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these values with OR. This parameter is optional.</p>
scored	<p>Whether the contact has been scored. The accepted Boolean values are:</p> <ul style="list-style-type: none">■ true■ false■ 1 (true)■ 0 (false) <p>This parameter is optional.</p>
silenceDuration	<p>Return only contacts where the duration (in microseconds) of recorded silence is equal to or greater than the value specified.</p> <div data-bbox="591 1430 1383 1619" style="border: 1px solid #ccc; border-radius: 10px; background-color: #e6f2ff; padding: 10px;"><p>Example: silenceDuration=5000 returns contacts where there are one or more instances of recorded silence equal to or greater than 5 seconds.</p></div> <p>This parameter is optional.</p>

Parameter	Description
silenceEvents	Return only contacts where the number of silence events are equal to or greater than the value specified. For example, silenceEvents=5 returns contacts where there are 5 or more silence events. This parameter is optional.
tagged	Whether the contact was tagged. The accepted Boolean values are: <ul style="list-style-type: none">■ true■ false■ 1 (true)■ 0 (false) This parameter is optional.
talkOverDuration	Return only contacts where the duration (in microseconds) of recorded talk over events are equal to or greater than the value specified. Example: talkOverDuration=5000 returns contacts where there are one or more instances of recorded talk over events equal to or greater than 5 seconds. This parameter is optional.
talkOverEvents	Return only contacts where the number of talk over events is equal to or greater than the value specified. Example: talkOverEvents=5 returns contacts where there are 5 or more talk over events. This parameter is optional.
team	A team's ID. The accepted value is integer ID. This parameter is optional.

Parameter	Description
training	<p>Whether the contact evaluation has been marked for training. The accepted Boolean values are:</p> <ul style="list-style-type: none"> ■ true ■ false ■ 1 (true) ■ 0 (false) <p>This parameter is optional.</p>
type	<p>The type of contact. The type parameter filters contacts based on upload states. The accepted values are:</p> <ul style="list-style-type: none"> ■ quality ■ archive <p>This parameter is optional. If you do not include this parameter, the query does not filter on upload states. Also note, the archive user role only has global scope when you specify the archive type.</p>

Examples

Combination search example:

The following request returns a list of all contacts recorded for quality purposes on or after 01/01/2009 (GMT).

```
GET ~/api/rest/recording/contact?beginTime=2009-01-01&type=quality
```

Searching for a metadata key example:

The following request returns a list of all contacts with the metadata called phone.

```
GET ~/api/rest/recording/contact?beginTime=2009-01-01&metadata=phone
```

The following returns a list of all contacts with the metadata called customerNo.

```
GET ~/api/rest/recording/contact?beginTime=2009-01-01&metadata=customerNo
```

Searching for a metadata key with a specific value:

The following request returns a list of all contacts with the metadata value of 555-1212.

```
GET ~/api/rest/recording/contact?beginTime=2009-01-01&metadata=phone~555-1212
```

Searching for silence events:

The following request returns a list contacts that contain two or more silence events of 10 seconds or greater.

```
GET  
~/api/rest/recording/contact?silenceEvents=2&silenceDuration=1000  
0
```

Sample Contact Object

```
{
  "hr" : false,
  "dnis" : "2111",
  "callDuration" : 15000,
  "ani" : "1581",
  "icmCallId" : "19291093",
  "assocCallId" : 290275383991253,
  "evalForm" : {
    "name" : "form1",
    "$ref" : "/api/rest/recording/evalform/5"
  },
  "team" : {
    "name" : "team1",
    "$ref" : "/api/rest/recording/team/1",
  "displayId" : "0.4"
  },
  "qualityReason" : {
    "text" : "First Call of Day",
    "reasonId" : 1,
    "key" : "rec_reason_what_first"
  },
  "agent" : {
    "lastName" : "Bunkowske",
    "username" : "bunkowm",
    "$ref" : "/api/rest/recording/person/1",
    "firstName" : "Mark",
    "displayId" : "0.1"
  },
  "evaluation" : {
    "stateId" : 1,
    "score" : 90,
    "$ref" : "/api/rest/recording/contact/1/eval/1"
  },
  "training" : false,
  "id" : 1,
  "startTime" : 1239308710000,
```

```
"tz" : "America/Chicago",
"recordingUrl" : "/api/rest/recording/contact/1/recording",
"audioUploaded" : true,
"archiveWF" : {
  "$ref" : "/api/rest/recording/workflow/265"
},
"group" : {
  "name" : "group1",
  "$ref" : "/api/rest/recording/group/1"
},
"evaluator" : {
  "lastName" : "Sillars",
  "username" : "sillarj",
  "$ref" : "/api/rest/recording/person/2",
  "firstName" : "Jay",
  "displayId" : "0.2"
},
"screenUploaded" : true,
"metadata" : {
  "$ref" : "/api/rest/recording/contact/1/metadata/"  },
"qualityWF" : {
  "$ref" : "/api/rest/recording/workflow/266"
}
}
```

Exporting Recordings

Use the POST method and the following resource URL to export a contact recording to a specified format.

```
POST /api/rest/recording/contact/{id}/export
```

Where {id} is the unique ID of the contact to export.

The Server API sends an alert to the user who initiated the export when the export completes or fails.

You format the request and response bodies using JavaScript Object Notation (JSON). The JSON object must include the `mediaFormat` attribute and one of the following values:

- WAV
- MP4

Sample Request Body

```
{"mediaFormat": "WAV"}
```

Sample Response

```
{  
  "id" : 271518269942,  
  "isComplete" : false,  
  "exportUrl" : "http://10.10.10.76/export/recording-1-  
271518269942.wav",  
  "mediaFormat" : "WAV"}  
}
```

Exporting Details

Use the GET method and the following resource URL to get details of an export.

```
/api/rest/recording/contact/{id}/export/{exportId}
```

Sample Response

```
{  
  "id" : 271518269942,  
  "isComplete" : true,  
  "exportUrl" : "http://10.10.10.76/export/ recording-1-  
271518269942.wav",  
  "mediaFormat" : "WAV"  
}  
{  
  "id" : 271518269942,
```

```
"isComplete" : false,
"error" : "Error converting audio.",
"exportUrl" : "http://10.10.10.76/export/ recording-1-
271518269942.wav",
"mediaFormat" : "WAV"
}
```

Downloading an Exported File

You can download an exported file by issuing the GET request to the URL specified by the exportURL attribute found in the response to the export requests when the export is complete. The server API deletes the exported file after you download it. If you do not download the exported file, the server API deletes the exported file after 24 hours.

Deleting a Recording

Use the DELETE method and the following resource URL to delete a recording.

```
/api/rest/recording/contact/{id}/recording
```

The DELETE method purges the recording the same way as the DB Cleaner service.

Editing Metadata

Use the PUT method and the following resource URLs to edit a recording's metadata.

- /api/rest/recording/contact/{id}/metadata
- /api/rest/recording/contact/{id}/metadata/{key}

The PUT method updates the state of the whole metadata set for a contact or you can specify a specific metadata in the metadata set to update.

If you provide a metadata key, the Server API only modifies the specified key and leaves the rest of the metadata associated with that contact alone.

Example: ~/api/rest/recording/contact/1/metadata/key

If you do not provide a metadata key, the Server API modifies the whole metadata set.

Example: `~/api/rest/recording/contact/1/metadata`

To delete a specified key in a metadata set, you must provide the key and set the value for that metadata key to null. The GET method returns the metadata for a contact.

Sample Request

```
{
  "myText" : {
    "value" : "my NEW sample data!"
  },
  "secretText" : {
    "value" : "I am a CHANGED value that is stored encrypted."
  }
}
```

Sample Response

```
{
  "myText" : {
    "encrypted" : false,
    "exportable" : true,
    "name" : "Sample Text Field",
    "value" : "my sample data!",
    "type" : "Text",
    "key" : "myText"
  },
  "secretText" : {
    "encrypted" : true,
    "exportable" : false,
    "name" : "Sample Secret Text",
    "value" : "I am a value that is stored encrypted.",
    "type" : "Text",
    "key" : "secretText"
  }
}
```

Exporting a Single Contact Example

1. Send a POST request to `/api/rest/recording/contact/1/export`. Include the following statement in the request body:

```
{"mediaFormat": "MP4"}
```

You should get the following request response:

```
{
  "id" : 303916344608,
  "isComplete" : false,
  "exportUrl" : "http://<server-ip>/export/recording-1-303916344608.mp4",
  "mediaFormat" : "MP4"
}
```

Note: If an error occurs during export, the request response includes an error attribute whose value describes the error and `isComplete` remains false.

2. To determine when the export completes, send the following GET request at regular intervals.

```
GET to /api/rest/recording/contact/1/export/303916344608
```

The response to the GET request appears in the same format as the response to the original POST request. Periodically repeat the GET request until the `isComplete` attribute is true or the error attribute has a value.

3. Send a GET request to the URL identified by the `exportUrl` attribute to download the exported file.

Example: `http://<server-ip>/export/recording-1-303916344608.mp4`

After you issue the request, the Server API transfers the exported file and then deletes the exported file from the server.

Recording Controls API

The Recording Controls API is a client API. The Recording Controls API provides a means for users to create an external application that interfaces with the Quality Management system and allows agents to perform the following actions:

- Tag calls for recording and retention
- Pause a recording
- Resume a recording
- Restart a recording

Note: The Restart command is not supported with Gateway Recording and will be removed in a future release.

- Delete calls marked for recording
- Attach user-defined metadata to calls
- Start and stop call segments
- Log on to phones and log out of phones—only Agent Recording supports this feature. This feature is not supported for Gateway Recording.
- Start and stop screen only recording

CAD and Finesse

You can integrate the Recording Controls API with the Cisco Agent Desktop (CAD) or Finesse using Hypertext Transfer Protocol (HTTP) actions. HTTP actions pass information in the form of an HTTP request from the agent desktop to a third-party application (in this case, the Recording Controls API) using HTTP methods. For more information on configuring Cisco Agent Desktop, see the *Quality Management Integration Guide for CAD and Finesse*.

Hot Desking

Hot Desking is a situation where one desk is shared between several people who use the desk at different times. This work surface can be an actual desk or just a terminal link. Companies use Hot Desking when not all the employees are in the office at the same time, or employees are not regularly in the office for very long.

The Recording Controls API provides a Login and Logout command to the Recording Cluster and Desktop Recording service for hot desking.

You must configure the device for hot desking. See “VoIP Devices” in the *Administrator Guide* for more information on hot desking.

Recording Controls API Requirements

Messages are sent to the Recording Controls applet on the Web Base server. Cisco Recording Controls listens on port 80 or 443 for incoming recording commands. The messages require the following information:

- Protocol: HTTP
- IP Address: <Web Base server>
- Port: 80 for HTTP and 443 for HTTPS
- Agent identifier—when sending API Recording commands to Recording Controls, you need to identify the Quality Management user associated with the command.

Agent Identifiers

The agent can be identified in one of the following ways:

- [Userdomain and Username](#)
- [Peripheral_id and Sender_id](#)

Userdomain and Username

You can use userdomain and username when you send an API command.

Example: `userdomain=CISCO&username=john.doe`

Peripheral_id and Sender_id

The `peripheral_id` and `sender_id` appear in the User ID column in Quality Management Administrator (Personnel > User Administration node). The User Administration window displays the agent identifier in the following format:

`<peripheral_id>.<sender_id>`

Example: `5000.1234`

You can use `sender_id` and `peripheral_id` when you send an API command.

Example: `sender_id=1234&peripheral_id=5000`

How to Find the Microsoft Windows Login Name

There are two ways to find a login name on a user's PC. The one you use depends on which application the user is using and the available API type.

- SENS events—If the application has access to the Windows API, you can use Windows events to get notification when a user logs in or logs out. The Desktop Recording service uses SENS events.
- Environment variables—When a user logs in, the following environment variables are set:
 - USERDOMAIN
 - USERNAME

Cisco Recording Controls uses environment variables.

Rules for Recording Controls Commands

- Commands are case-insensitive.
- You can send multiple commands for the same call.

Example: You can attach metadata to a call and tag the same call for retention. However, once you delete a call using the delete command, the metadata and tag commands have no effect.

- In a multi-tenancy environment, the `sender_id` and `extension` must be unique across the entire system.
- The following list displays the valid recording commands that you can enter in the `recordingcontrols.properties` file. The recording commands a Recording Controls user can see and use are the ones that appear in the `recordingcontrols.properties` file.
 - `record`
 - `pause`
 - `resume`
 - `restart`

Note: The Restart command is not supported with Gateway Recording and will be removed in a future release.

Recording and will be removed in a future release.

- `delete`
- `login`
- `logout`

Note: The Recording Controls IP Phone Service does not support the Login and Logout commands.

- `metadata`
 - `start`
 - `stop`
 - `start_screen`
 - `stop_screen`
- Recording Controls ignores all other words.

Do not enter spaces between the words on these lines. Use commas as separators between the commands.

- You can control the Recording Controls commands available to Quality Management users who are agents and knowledge workers.

Example: You could assign record, pause, resume, restart, delete, start, and stop to agents and assign metadata to knowledge workers.

```
recordingcontrols.agent=record,pause,resume,restart,delete,login,logout,start,stop
recordingcontrols.know=metadata
```

- The order in which commands appear on these lines determines the order in which they appear in the Recording Controls IP Phone Service. The command order does not apply to the Recording Controls Browser application.
- If you remove commands from these lines, the commands no longer appear in either the Recording Controls IP Phone Service or Recording Controls Browser application.

Example: When you remove a command in the Recording Controls Browser application, the button no longer appears and the remaining buttons align themselves to fill the space.

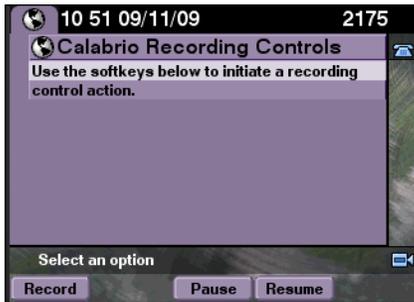
- The Recording Controls IP Phone Service can display up to four soft buttons at a time (some phones, like the IP Communicator soft phone, can allow as many as five buttons).

To see additional commands, the user must press a button to display more commands. For this reason, you should place the most frequently used commands at the beginning of the list. This makes command navigation easier.

Example: If you want a blank button to appear on the IP phone service screen, you can simply omit the command name in the string, using two consecutive commas, as follows:

```
recordingcontrols.agent=record,,pause, resume
```

When you enter a string with two consecutive commas, the result on the main page of IP phone service looks like the following figure.



The result on the Recording Controls Browser application looks like the following figure.



For more information about the Recording Controls Browser application, see the *Recording Controls User Guide*.

- If you are using Gateway Recording without reconciliation Gateway Recording, none of the API commands are supported.
- If you are using Gateway Recording with reconciliation, the following Recording Controls API commands are not supported:
 - Login
 - Logout
 - Start Segment
 - Stop Segment
 - Start Screen
 - Stop Screen

- These commands are generally issued at the time of recording. Gateway Recording only supports commands that can be issued after the call is recorded.
- For Gateway Recording, the use of recording commands is not supported for the extensions in the exclusion list.

Recording Commands

This section explains the following concepts:

- Syntax of recording commands
- Function of each recording command
- Active and last calls
- Using commands with an outbound dialer

Command Syntax

The Recording Controls API supports the following HTTP methods:

- GET

```
http://<Web Base server
IP>/recordingcontrols/rest/<command>?<agent
identifier>&<variable>=<variable value>
```

- POST

```
http://<Web Base server IP>/recordingcontrols/rest/<command>
{
  "<agent identifier>": "<agent identifier>",
  "metadata": {
    "<variable>": "<variable value>"
  }
}
```

where:

- <Web Base server IP> is the IP address of the Web Base server.
- <command> is the recording controls command you want to send. Valid

commands are record, pause, resume, restart, delete, login, logout, metadata, start, stop, start_screen, and stop_screen.

- <agent identifier> is the peripheral_id and sender_id or the username or userdomain of the agent. For more information on agent identifiers, see [Agent Identifiers](#).
- <variable> and <variable value> (optional) are additional information you want to attach to the command.

Example: <key>=<value> or <key>:<value>

Commands that require variables are login, logout, metadata, start_screen, and stop_screen.

Command Functions

Recording commands allow you to control a recording.

Example: You can use recording commands to record a call, pause the recording, and attach metadata to a recording.

The following table describes how the recording commands interact with each other and the Quality Management components.

Recording commands

Command	Function
Record	<p>Records a call and uploads the call to the Quality Management server at the end of the day.</p> <p>In the Recording Controls API, the <command> is record.</p> <p>The Record Tag command behaves as follows:</p> <ul style="list-style-type: none"> ■ Agent Recording—marks a call for recording, even if archiving is not enabled and the call does not meet the workflow criteria. The Record Tag command overrides both the Don't Record list and the workflow classifiers. ■ Gateway/MediaSense Recording—marks a recording as tagged if archiving is enabled and the call meets the workflow criteria. The Record Tag overrides the workflow, but does not override an exclusion list in the Exclusion List window because the root contact does not know the agent's identity when recording. See "Recording Controls Considerations for Gateway Recording" in the <i>Administrator Guide</i> for more information. <p>Quality Management stores agent-tagged calls with the Agent Tagged reason code, and saves them for the retention time configured in Quality Management Administrator.</p> <p>Agent Recording:</p> <ul style="list-style-type: none"> ■ The Record Tag command is valid for the active call and the last call. ■ If Quality Management is not recording the active call, Quality Management starts recording the call when you invoke the command and adds the Agent Tagged reason code. ■ If Quality Management is recording two active calls (for example, an inbound ACD call and an outbound consultation call), Quality Management tags the call that triggered the recording to begin. ■ If Quality Management is not recording two active calls (for example, an inbound ACD call and an outbound consultation call), Quality Management begins recording the first call sent to

Command	Function
	<p>the agent, based on the call start times, and tags the first call when you invoke the Record Tag command.</p> <p>Gateway/MediaSense Recording and Agent Recording:</p> <ul style="list-style-type: none">■ When Quality Management actively recording a call, the Tag command adds the Agent Tagged reason code to the data associated with the call.■ When Quality Management is not actively recording a call, the Tag command changes the reason code associated with the last recorded call to the Agent Tagged reason.■ If Quality Management did not record the last call, nothing happens. Quality Management cannot update the reason code when no recording is available.

Command	Function
Pause	<p>Temporarily halts the recording of:</p> <ul style="list-style-type: none"> ■ Audio—the audio recording is silent for the duration of the pause in the final recording where an agent discussed sensitive information. ■ Screen—the screen recording displays the following message for the duration of the pause in the final recording where an agent typed sensitive information on the screen. <pre style="margin-left: 40px;">Screen recording paused</pre> <p>When you cannot record sensitive information (such as Social Security numbers) for security or liability reasons, use the Pause command. The Pause command allows you to omit sensitive information from the final recording. This command adheres to the Payment Card Industry Data Security Standard (PCI DSS) for protecting consumer data.</p> <p>Calls are available for playback prior to reconciliation with silence where an agent used the Pause command. These calls are accessible by anyone with the archive user role.</p> <p>When using the Pause command, note the following:</p> <ul style="list-style-type: none"> ■ Agent Recording: <ul style="list-style-type: none"> • The pause command is valid for active calls only. • If you send a pause command for a call currently in the paused state, the pause command has no effect. • The pause command does not affect live monitoring. ■ Gateway/MediaSense Recording delays the pause. The pause will appear in the recording after the recording is uploaded. <p>Issue the Resume command when you want to start recording after a pause.</p>

Command	Function
Resume	<p data-bbox="586 279 1349 352">Resumes recording after you issued a Pause command to stop the recording.</p> <p data-bbox="586 380 786 411">Agent Recording:</p> <ul data-bbox="618 438 1365 831" style="list-style-type: none"><li data-bbox="618 438 1321 470">■ The Resume command affects voice and screen recording.<li data-bbox="618 497 1338 571">■ If the call is not currently paused, the Resume command has no effect.<li data-bbox="618 598 1235 630">■ The Resume command is valid for active calls only.<li data-bbox="618 657 1365 730">■ If you do not use the Resume command, the point at which you paused the recording is the end of the audio recording.<li data-bbox="618 758 1305 831">■ A Resume command does not appear as a mutual silence event or talkover event during post-call processing.

Command	Function
Restart	<p data-bbox="586 279 1040 310">Restarts or starts the recording of a call.</p> <div data-bbox="586 331 1382 478" style="background-color: #e6f2e6; border: 1px solid #ccc; padding: 10px;"><p data-bbox="630 367 1308 443">Note: The Restart command is not supported with Gateway Recording and will be removed in a future release.</p></div> <p data-bbox="586 499 1243 531">In the Recording Controls API, the <command> is restart.</p> <p data-bbox="586 558 786 590">Agent Recording:</p> <ul data-bbox="621 617 1360 1121" style="list-style-type: none"><li data-bbox="621 617 1360 779">• If Quality Management is currently recording an active call, the Restart command stops the audio and screen recording, deletes that recording, and restarts recording the call from the point when you issued the Restart command.<li data-bbox="621 806 1360 879">• If Quality Management is not currently recording an active call, the Restart command starts audio and screen recording.<li data-bbox="621 907 1224 938">• The Restart command is valid for active calls only.<li data-bbox="621 966 1360 1121">• Quality Management assigns an Agent Tagged reason code to calls recorded using the Restart command. Quality Management saves the agent tagged calls even if archiving is not enabled and the call does not meet workflow criteria. <p data-bbox="586 1150 1297 1224">Gateway/MediaSense Recording does not support the Restart command.</p> <p data-bbox="586 1253 1365 1455">Use this command if you call someone and you are immediately placed on hold for a long time. Issue the Restart command when you leave the hold queue and begin speaking to a person. This eliminates the period when you are on hold (for example, 20 minutes of recorded on-hold music).</p>

Command	Function
Delete	<p>Marks a recording for deletion, even if archiving is enabled, the call meets workflow criteria, the extension is in the inclusion list, or it is tagged for retention. The Delete command deletes the recorded files and any metadata, and uploads the basic contact data to Quality Management to maintain accurate call counts.</p> <p>In the Recording Controls API, the <command> is delete.</p> <ul style="list-style-type: none"> ■ The Delete command is valid for the active call only. ■ The Delete command has precedence over all other commands. ■ Once you delete a call you cannot record it by issuing the Record Tag command. ■ Deleted calls are not available for archive purposes or quality management purposes. ■ You cannot view deleted calls in Unified Workforce Optimization. ■ For Gateway/MediaSense Recording, the recording is deleted for the person who sends the command, but the audio recording might continue to exist in the root call or in other calls associated with this call.
Login	<p>Sends a login request that associates an agent with the specific extension for hot desking.</p> <p>The Recording Controls IP Phone Service does not have login/logout capabilities. Use Cisco's Extension Mobility IP Phone application to log in by phone.</p> <p>In the Recording Controls API, the <command> is login. You must include the unique extension of the phone that the agent is logging into.</p> <p>This command is not supported if you are using Gateway/MediaSense Recording.</p>

Command	Function
Logout	<p>Sends a logout request that associates an agent with the specific extension for hot desking.</p> <p>In the Recording Controls API, the <command> is logout.</p> <p>This command is not supported if you are using Gateway/MediaSense Recording.</p>

Command	Function
Metadata	<p>The Metadata command attaches metadata to an active call. If Quality Management does not upload the current call (or previous) for archiving because of workflow criteria, then the metadata will be uploaded to the database but will not appear in the interface.</p> <p>In the Recording Controls API, the <command> is metadata. You must include at least one key/value pair (<key>=<value> or <key>:<value>).</p> <ul style="list-style-type: none">■ The Metadata command is valid for the active call and the last call.■ You can associate maximum of 30 metadata items with a call. You can accomplish this with 30 Metadata commands containing one key/value pair each, or one Metadata command containing up to 30 key/value pairs.■ You can only attach metadata defined in Quality Management Administrator (Recordings > Metadata) to a call. If you add an unknown key to a Metadata command, Quality Management ignores the unknown key. <p>The Metadata command interacts with the active call, including the time up until the next call starts. If you invoke the Metadata command during a call, Quality Management uploads the metadata to the database at the same time as the rest of the call data. If you invoke the Metadata command after the call but before the next call, Quality Management uploads the metadata separately at the time you invoke the command and Quality Management stores the metadata with the last known call. Calls that occur after a recorded call that do not match the inclusion list are not counted as the next call.</p> <div data-bbox="591 1493 1382 1766" style="background-color: #e6f2e6; padding: 10px;"><p>Note: Quality Management resets the last known call at login, so Quality Management cannot attach metadata to the last known call before logout or shutdown after the next login occurs. Quality Management attaches metadata to calls that span the configured end of day/upload time.</p></div>

Command	Function
	<p>Successive calls to the Metadata command using the same key name update the existing metadata for that call.</p> <p>Specifying an empty value for a key removes that metadata field association for the call.</p> <p>Valid formats for metadata are as follows.</p> <ul style="list-style-type: none">■ Dates—Dates must be in yyyy-mm-dd format (for example 2009-09-24).■ Numbers—Numbers can start with and contain a decimal point (for example, valid numbers are .30, 10.7, and 2500). Numbers cannot end with a decimal point or contain a comma (for example, invalid numbers are 30. and 2,500).■ Text—Text key values cannot contain the reserved characters. <div data-bbox="651 884 1365 982" style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; background-color: #e6f2ff;"><p>Example: & or =</p></div> <p>All other alphanumeric characters are valid.</p> <p>You can find the decimal point in the * key menu and the dash in the zero key menu on your phone.</p>

Command	Function
Start Segment	<p data-bbox="586 279 1382 394">Starts the audio and screen recording of an active call. This command allows you to override the automatic exclusion lists to start recording their current call and treat it as a normal contact.</p> <p data-bbox="586 426 1219 453">In the Recording Controls API, the <command> is start.</p> <p data-bbox="586 485 786 512">Agent Recording:</p> <ul data-bbox="618 543 1370 974" style="list-style-type: none"><li data-bbox="618 543 1370 613">■ If Quality Management is not currently recording an active call, the Start Segment command starts audio and screen recording.<li data-bbox="618 644 1370 714">■ If Quality Management is currently recording an active call, the Start Segment command has no effect.<li data-bbox="618 745 1370 814">■ If the active call ends before the recording is stopped by the agent, the recording is saved according to workflow criteria.<li data-bbox="618 846 1370 873">■ The Start Segment command does not override the workflow.<li data-bbox="618 905 1370 974">■ The Start Segment command does override the exclusion list in the Exclusion List window. <div data-bbox="654 999 1365 1360" style="border: 1px solid #ccc; padding: 10px; background-color: #e6f2ff;"><p data-bbox="691 1035 1328 1325">Example: If you are using an Outbound Dialer, you can add the Outbound Dialer to the exclusion list to prevent recording from starting when an agent logs in. The agent can use the Start Segment and Stop Segment commands to override the exclusion list and record each outbound call. For more information, refer to Using Commands with an Outbound Dialer.</p></div> <p data-bbox="586 1383 1377 1453">Gateway/MediaSense Recording does not support the Start Segment command.</p>

Command	Function
Stop Segment	<p data-bbox="586 279 1365 352">Stops the audio and screen recording of an active call. The recording is then saved according to workflow criteria as a new contact.</p> <p data-bbox="586 380 1219 411">In the Recording Controls API, the <command> is stop.</p> <ul data-bbox="618 438 1333 615" style="list-style-type: none"><li data-bbox="618 438 1333 512">■ Agent Recording supports the Stop Segment command only during active calls.<li data-bbox="618 539 1333 615">■ Gateway/MediaSense Recording does not support the Stop Segment command. <p data-bbox="586 642 1382 758">The agent can use the Stop Segment command to stop the recording after a sale has been made and before payment information is taken in order to omit customer data in adherence with PCI DSS.</p>

Command	Function
Start Screen	<p>Starts screen recording regardless of whether or not you are participating in an active call. Use this command to record chat or email interactions with a customer.</p> <p>Voice contact recordings and screen only contact recordings can be bracketed or interleaved.</p> <div data-bbox="591 520 1385 926" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p>Example: You can send the Start Screen command to record your screen while not participating in an active call. If you receive a phone call or make a call during this time, a separate voice and screen contact might be created according to workflow (or you can send the Start Segment and Stop Segment commands to create the contact). After the active call has ended, another screen only contact is created and will continue until you send the Stop Screen command.</p> </div> <p>In the Recording Controls API, the <command> is start_screen.</p> <p>Agent Recording:</p> <ul style="list-style-type: none"> ■ The Start Screen command is only supported with the Advanced bundle. ■ If Quality Management is not currently recording an active call, the Start Screen command starts screen only recording. ■ If Quality Management is currently recording an active call, the Start Screen command has no effect on the current recording. The screen only recording will begin after the active call has ended (if the Stop Screen command has not been issued). The call recording and the screen only recording are saved as separate contact recordings. <p>Gateway/MediaSense Recording does not support the Start Screen command.</p> <p>After issuing the Start Screen command, you can send other commands. The following list contains the commands that are supported with screen only recording and the expected behavior.</p>

Command	Function
	<ul style="list-style-type: none">■ Pause—Pauses the current screen only recording■ Resume—When sent after the Pause command, resumes the screen only recording■ Restart—Stops the screen only recording, deletes that recording, and restarts the screen only recording from the point when you issued the Restart command. <div data-bbox="652 577 1367 764" style="border: 1px solid black; background-color: #e0f0e0; padding: 5px;"><p>Note: The Restart command is not supported with Gateway Recording and will be removed in a future release.</p></div> <ul style="list-style-type: none">■ Delete—Deletes the screen only recording. You must send the Stop Screen command either before or after the Delete command in order to be able to send the Start Screen command again to start another screen only recording.■ Metadata—Attaches metadata to the active screen only recording <p>All other commands have no affect on the current screen only recording.</p> <p>Issue the Stop Screen command to stop screen only recording.</p>

Command	Function
Stop Screen	<p>Stops screen recording.</p> <p>In the Recording Controls API, the <command> is stop_screen.</p> <p>Agent Recording:</p> <ul style="list-style-type: none"> ■ The Stop Screen command is only supported with the Advanced bundle. ■ If Quality Management is currently recording screen only, the Stop Screen command stops the screen recording. The Stop Screen command only has an effect if you previously issued the Start Screen command. ■ If the Stop Screen command is not sent after the Start Screen command, the maximum contact recording length is 4 hours. <p>Gateway/MediaSense Recording does not support the Stop Screen command.</p>
Config	<p>Displays configuration information for the specified user. A user is configured for desktop recording if no serverHost is returned. In this case you need to send commands via the applet, not the server.</p>

Effects of Issuing Recording Control Commands

The following table indicates the effect of issuing a recording command when the call is currently being recorded, and when the call is not being recorded.

Effect of issuing recording control commands on calls

Command	Currently Recording	Not Currently Recording
Record	Sets Reason Code to TAGGED.	Call recording starts. Sets Reason Code to TAGGED.
Pause	Recording paused.	None.
Pause URL	Recording paused.	None.
Resume	Recording resumes (if previously paused).	None.

Command	Currently Recording	Not Currently Recording
Restart	Recording restarts. Sets Reason Code to TAGGED.	Recording starts. Sets Reason Code to TAGGED.
Delete	Call deleted.	None.
Start Segment	None.	Recording starts.
Stop Segment	Recording stops and is saved according to workflow criteria.	None.
Start Screen	If there is a current voice and screen recording, the Start Screen command has no effect on the current recording. The screen only recording will begin after the active call has ended (if the Stop Screen command has not been issued) and will be created as a separate contact recording. If there is a current screen only recording, the Start Screen command has no effect.	Screen only recording starts.
Stop Screen	If there is a current screen only recording, screen recording stops. The Stop Screen command only has an effect if you previously issued the Start Screen command.	None.

Active and Last Call

You must understand the difference between the terms *active call* and *last call*. Some commands can apply to either one of these call types. Some commands can apply to a single call type.

An active call occurs when the Quality Management user is on a call with one or more parties. A call on hold is still an active call. The active call starts when the Quality Management user receives the call (phone is ringing) or makes a new call. The active call ends when the user hangs up the phone.

The last call is the previously recorded call. Any valid recording commands sent after a call ends, and until another call, that matched the inclusion list, is received or made by the user, apply to the last call.

The following table indicates whether the recording command applies to the active call, the last call, or both.

Recording commands that support Active Calls or Last Calls

Command	Active Call	Last Call
Pause	Yes	No
Pause URL	Yes	No
Resume	Yes	No
Record	Yes	No
Restart	Yes	No
Delete	Yes	No
Login	No—The recording command applies to the next active call.	No
Logout	Yes—An active call stop recording.	No
Metadata	Yes	Yes
Metadata and &active_call_only=y=true	Yes	No
Start Segment	Yes	No
Stop Segment	Yes	No
Start Screen	Yes—And when there is no active call.	No
Stop Screen	Yes—And when there is no active call.	No

Command Examples

Login Command Examples

- GET

```
http://<Web Base server IP>  
/recordingcontrols/rest/login?sender_id=1234&peripheral_  
id=5000&extension=1234
```

- POST

```
http://<Web Base server IP>/recordingcontrols/rest/login  
{  
  "sender_id": "1234"  
  "peripheral_id": "5000"  
  "extension": "1234"  
}
```

Metadata Command Examples

- GET

```
http://<Web Base server IP>  
/recordingcontrols/rest/metadata?userdomain=acme&username=jan  
edoe&<key>=<value>&<key>=<value>
```

- POST

```
http://<Web Base server IP>/recordingcontrols/rest/metadata  
{  
  "userdomain": "acme"  
  "username": "janedoe"  
  "metadata": {  
    "<key>": "<value>",  
    "<key>": "<value>"  
  }  
}
```

Pause Command Examples

- GET

```
http://<Web Base server  
IP>/recordingcontrols/rest/pause?sender_id=1234&peripheral_  
id=5000
```

- POST

```
http://<Web Base server IP>/recordingcontrols/rest/pause  
{  
  "sender_id": "1234"  
  "peripheral_id": "5000"  
}
```

Resume Command Examples

- GET

```
http://<Web Base server IP>  
/recordingcontrols/rest/resume?userdomain=acme&username=janed  
oe
```

- POST

```
http://<Web Base server IP>  
/recordingcontrols/rest/resume  
{  
  "userdomain": "acme"  
  "username": "janedoe"  
}
```

Start Segment Command Examples

- GET

```
http://<Web Base server  
IP>/recordingcontrols/rest/start?sender_id=1234&peripheral_  
id=5000
```

- POST

```
http://<Web Base server IP>/recordingcontrols/rest/start  
{  
  "sender_id": "1234"  
  "peripheral_id": "5000"  
}
```

Using Commands with an Outbound Dialer

An outbound dialer creates a single “nailed up” call for the entire time you are logged in. This results in all of your outbound calls being combined into one large recording, even though you might make numerous outbound calls during your session. Use the Start Segment and Stop Segment commands to break this large nailed up call into multiple contact recordings.

Send the Start Segment command at the beginning and the Stop Segment command at the end of each outbound call to create a unique contact recording for each outbound call. The Start Segment command starts the audio and screen recording of an active call. The Stop Segment command stops the recording. The recording is then saved according to workflow criteria as a new contact. Send the Start Segment command again at the beginning of your next outbound call to start another unique contact recording.

You can also add the Outbound Dialer to the exclusion list to prevent recording from starting when an agent logs in. The agent can use the Start Segment and Stop Segment commands to override the exclusion list and record each outbound call.

For information about commands, refer to [Command Functions](#).

Integrating Recording Commands with CAD and Finesse

For more information on integrating CAD and Finesse with the Recording Controls API, refer to the *Quality Management Integration Guide for CAD and Finesse*.

Configuring Recording Controls

Cisco Recording Controls is installed on the Quality Management Web Base server when you install Quality Management.

Recording Controls software has a configuration file called `recordingcontrols.properties`. This configuration file controls the behavior of the Recording Controls browser application and user applications.

This file resides on the Quality Management Web Base server where you installed the Recording Controls browser application. The default folder is one of the following:

```
C:\Program Files\Cisco\WFO_QM\config
```

The following example displays a typical `recordingcontrols.properties` file.

```
#log4j.rootLogger=INFO, LOG, DBG log4j.rootLogger=DEBUG, LOG, DBG
#log4j.rootLogger=CALL#com.cisco.util.log.SplkLevel, LOG, DBG
#log4j.rootLogger=TRACE, LOG, DBG
#log4j.rootLogger=DUMP#com.cisco.util.log.SplkLevel, LOG, DBG

log4j.appender.LOG=com.cisco.util.log.SplkRollingFileAppender
log4j.appender.LOG.layout=org.apache.log4j.PatternLayout
log4j.appender.LOG.Threshold=INFO#com.cisco.util.log.SplkLevel
log4j.appender.LOG.File=../log/recordingcontrols.loglog4j.appende
r.LOG.MaxFileSize=3MB log4j.appender.LOG.MaxBackupIndex=2
log4j.appender.LOG.layout.ConversionPattern=%d %-5p %X{EC}%m%n

log4j.appender.DBG=com.cisco.util.log.SplkRollingFileAppender
log4j.appender.DBG.layout=org.apache.log4j.PatternLayout
log4j.appender.DBG.Threshold=DUMP#com.cisco.util.log.SplkLevel
log4j.appender.DBG.File=../log/recordingcontrols.dbglog4j.appende
r.DBG.MaxFileSize=10MB log4j.appender.DBG.MaxBackupIndex=20
```

```
log4j.appender.DBG.layout.ConversionPattern=%d %-5p %X{EC} [%t|%X  
{CML}] %m%n
```

```
splk4j.appender.DBG.accept=STACK#com.cisco.util.log.SplkLevel  
splk4j.watch.check.sec=5 splk4j.watch.error.sec=600
```

```
recordingcontrols.agent=record,pause,resume,restart,delete,metada  
ta,login,logout
```

```
recordingcontrols.know=record,pause,resume,restart,delete,metadat  
a recordingcontrols.title=Cisco Recording Controls
```

```
recordingcontrols.pauseurl=http://www.abc.com/us
```

Use the `recordingcontrols.properties` file to:

- Control the debug levels for the Recording Controls browser application
- Change the title that appears at the top of the browser application and IP Phone service
- Control the Recording Controls buttons available to Quality Management users who are agents and knowledge workers
- Change the order in which the Recording Controls buttons appear in the IP Phone service
- Specify the URL used when pausing and resuming a screen recording

For more information on files with the `PROPERTIES` extension, see the *Quality Management Troubleshooting Guide*.

Changing the Debugging Level

The first 25 lines in the `recordingcontrols.properties` file start with “log4j” or “splk4j.” These lines control the type and amount of debugging information generated by the Recording Controls webapp when it is running. This topic explains how to change debugging levels in the properties file.

See “Logs and Debugging” in the *Quality Management Troubleshooting Guide* for additional debugging information.

Pausing Audio and Screen Recording

Recording Controls allows the agent to pause audio and screen recording while the agent is working on a pop-up browser window or browser tab.

Example: You might decide that you do not want to include a credit card number or social security number in a recording when the agent goes to a specific URL.

You have two configuration options for pausing a recording while working on another pop-up browser window or browser tab:

- Single pause URL—if you only want to pause recording on one URL, you can configure that URL in the `recordingcontrols.properties` file, and provide a pause link for that page to your agents.
- Multiple pause URLs—if you want to pause recording on more than one URL, you can provide a pause URL link for each URL to your agents.

Recording Controls pauses the recording when the agent enters the Pause URL command in their web browser. The URL specified in the Pause URL command is displayed in a pop-up browser window and Recording Controls pauses recording. Recording resumes when the agent closes the pop-up browser window.

Single Pause URL

The format for specifying a Pause URL in the `recordingcontrols.properties` file is:

```
recordingcontrols.pauseurl=<URL>
```

Where `<URL>` is the web address you want to use.

Example: `http://www.cisco.com`

You can only specify one URL in the `recordingcontrols.properties` file.

While on a call, the agent must use the following URL format to pause the recording when visiting the Pause URL specified in the `recordingcontrols.properties` file.

```
http://<Web Base server>/recordingcontrols/pause.html
```

Where <Web Base server> is the IP address of the Quality Management Web Base server.

When this Pause URL is entered into a web browser, a pop-up browser window or browser tab appears displaying the web address that was specified in the recordingcontrols.properties files, and Recording Controls pauses the recording. When the pop-up window or browser tab is closed, the recording resumes.

Example: If the URL specified in the recordingcontrols.properties file is www.cisco.com, then Recording Controls stops recording when you enter the Pause URL command (`http://<Web Base server>/recordingcontrols/pause.html`), and www.cisco.com appears in the pop-up browser window or browser tab.

Multiple Pause URLs

If you want agents to access multiple Pause URLs, there is no need to specify a Pause URL in the recordingcontrols.properties file.

While on a call, the agent must use the following format for each URL to pause the recording:

```
http://<Web Base server>/recordingcontrols/pause.html?url=<URL>
```

Where <Web Base server> is the IP address of the Quality Management Web Base server and <URL> is the web address you want to use.

Example: `http://www.cisco.com`

If you want your agents to use the Pause URLs for www.cisco.com and www.acme.com, the format for each Pause URL command is as follows:

Example:

```
http://<Web Base  
server>/recordingcontrols/pause.html?url=http://www.  
cisco.com
```

```
http://<Web Base  
server>/recordingcontrols/pause.html?url=http://www.acme.com
```

Assigning Pause URLs to Agents

After you create one or more Pause URLs, you need to send the Pause URLs to your agents and tell the agents to:

- Bookmark each Pause URL so they have the Pause URLs when they need to pause a recording
- Always allow pop-ups for each Pause URL in the web browser
- Use the Pause URLs to open a pop-up browser window to a specific website and pause recording
- Close the pop-up browser window or browser tab when you are done entering information to resume recording

Note: Recording does not resume until you close the pop-up browser window or browser tab.

Changing the Title

You can change the title that appears at the top of the Browser application or IP Phone service in the recordingcontrols.properties file.

Example: You could change the title to Acme Recording Controls.

```
recordingcontrols.title=Acme Recording Controls
```

Configuring the IP Phone Service

The IP Phone service runs as a Phone XML application on Cisco phones. This section describes how to configure Unified CM for the IP Phone service.

After you configure the IP phone service for Recording Controls IP Phone Service in Unified CM (see [Configuring Unified CM for IP Phone Service](#)) and

assign the IP phone service to the Quality Management users' IP phones, they can access it just like any other IP phone service by pressing the Services button on their phone.

Recording Controls IP Phone Service Considerations

When configuring Recording Controls for the IP Phone service, consider the following:

- The Recording Controls IP Phone service only runs in a Cisco environment.
- The Recording Controls IP Phone service only supports Network Recording and Server Recording (SPAN). If a user configured for Desktop Recording (Endpoint) tries to access the IP Phone service, an error appears.
- To use the Recording Controls IP Phone service, you must configure an IP phone service and assign agents to the IP phone service in Cisco Unified CM.
- The Recording Controls IP Phone Service supports all Cisco IP phones that can support services, as well as the Cisco IP Communicator soft phone.
- The Recording Controls IP Phone Service does not have login/logout capabilities. Use Cisco's Extension Mobility IP Phone application to log in by phone.

Configuring Unified CM for IP Phone Service

Before you can use the Recording Controls IP Phone service, you must configure an IP phone service and assign the service to Quality Management users' phones in Unified CM to support the Recording Controls IP Phone Service. The information provided in this section applies to Unified CM 5.1. Other Unified CM versions might differ. Please refer to the appropriate Unified CM Administration documentation for your version of Unified CM for the most accurate information.

Regardless of the Unified CM version, there are two basic steps required to create an IP phone service:

1. Create an IP phone service definition with a name and URL.
2. Assign the IP phone service to one or more IP phones.

Creating an IP Phone Service Definition

1. Log into Cisco Unified CM Administration.
2. From Cisco Unified CM Administration, choose Device > Device Settings > Phone Services. The Find and List IP Phone Services window appears.
The Find and List IP Phone Services window appears.
3. Click Add New. The IP Phone Services Configuration window appears.
4. Enter the information in the Service Information pane for the Recording Controls IP Phone Service, select the Enable check box, and click Save.

When you enter information in these fields, note the following:

- You can assign any name you want to the Recording Controls IP Phone Service in the Service Name field. You can also assign the same name to the ASCII Service Name and Service Description fields. This name appears on the user's phone when the user presses the Services button on the phone. In this document, the examples use Recording Control as the service name.
- You must enter the Service URL using the following format:

```
http://<IP address>/recordingcontrols/ipp/main
```

Where <IP address> is the IP address or hostname for the Quality Management Web Base server.

- Choose XML Service as the Service Category.
- Choose Standard IP Phone Service as the Service Type.

The Service Parameter Information pane appears on the IP Phone Services Configuration window. No additional parameters or information is required.

Assigning the IP Phone Service to Agents' Phones

The agent phones must comply with the following requirements before you can perform this task.

- The phone must be associated with a Recording Cluster in the VoIP Device table in Quality Management Administrator
- The agent must be assigned to the phone, or logged into the phone with a configured Extension Mobility (EM) profile in Quality Management Administrator
- The Quality Management Web Base server must be able to open the IP phone's configuration page (<http://<Device IP>/DeviceInformationX>)

This task shows you how to assign the IP phone service to Quality Management agent phones. The agent phones configured in this step are the phones that can use the Recording Controls IP Phone Service.

1. From Cisco Unified CM Administration, choose Device > Phone.

The Find and List IP Phone Services window appears.

2. Use the search options to locate the phone you want to assign the IP phone service to.
3. Choose the phone you want from the Search Results list.

The Phone Configuration window appears.

4. Choose Subscribe/Unsubscribe Services from the Related Links drop-down list, and then click Go.

The Subscribed Cisco IP Phone Services dialog box appears.

5. Choose the service you created in from the Select a Service drop-down list, and then click Next.

The Subscribed Cisco IP Phone Services window displays the information associated with the selected service.

6. Click Next.

7. Click Subscribe to add the service to the list of services assigned to the agent's phone.

The Subscribed Cisco IP Phone Services dialog box displays all subscribed services.

8. Click Save.

The new IP phone service appears in the service list when the agent presses the Services button on their hard or soft phone.

9. Repeat steps 1-8 for each Quality Management agent you want to assign

this service to.

The Cisco IP Phone Service can only be used by Quality Management agents who are using the Network Recording service.

Verifying that the Recording Controls IP Phone Service is Working

If you are using the Recording Controls IP phone service, verify the service is configured correctly.

Only users who are properly configured to use Network Recording service can use the Recording Controls IP Phone Service. If you configure a user to use Desktop Recording service, they must use the Recording Controls Browser application to control their recordings.

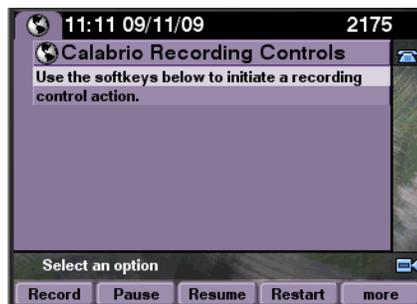
1. From your Cisco IP phone or Cisco IP Communicator soft phone, press the Services button. The Services menu appears.



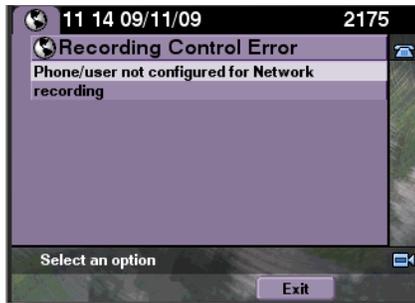
2. Select the IP phone service for the Recording Controls.

You can assign any name to this service. In this example, the name of Recording Controls IP Phone Service is Recording Control.

If the Recording Controls IP Phone Service is working, the Recording Controls IP Phone Service base screen appears.



If the Quality Management user is not configured correctly for Network Recording, the Recording Controls IP Phone Service displays an error message.



3. If this error message appears, check the user's configuration settings in Quality Management Administrator and try again.

Recording Verification API

The Recording Verification API is a client API. The Recording Verification API provides a means for users to create an external application that interfaces with the Quality Management system and allows users to check the call status and verify it is being recorded.

The Recording Verification API allows you to search call status by user name. You can also include the Automatic Number Identification (ANI) and/or Dialed Number Identification Service (DNIS) in your search.

Cisco Recording Verification is designed to work with a custom CTIOS application, called HUVR, that is used for outbound dialing recording verification.

The Recording Verification API is part of Cisco Recording Controls and is installed when Cisco Recording Controls is installed. This section contains only information on Recording Verification.

See [Recording Controls API](#) for more information.

Recording Verification API Requirements

Your site must be configured for Agent Recording. Gateway Recording is not supported. For more information on Agent Recording, see the *Quality Management Design Guide*.

Messages are sent to the Recording Controls on the Web Base server. Cisco Recording Controls listens on port 80 or 443 for incoming recording commands. The messages require the following information:

- Protocol: HTTP
- IP Address: <Web Base server>
- Port: 80 for HTTP and 443 for HTTPS

Recording Verification Commands

Messages are sent to the Recording Controls applet on the Web Base server. By default, Cisco Recording Controls listens on port 80 or 443 for incoming recording commands. The messages require the following information:

- Protocol: HTTP
- IP Address: <Web Base server>
- Port: 80 if you are not using HTTPS and 443 if you are using HTTPS

Command Syntax

The Recording Verification API supports the following HTTP GET method:

```
http://<Web Base server IP>/recordingcontrols/rest/call_
status?username=<first name>.<last name>&ani=<variable
value>&dnis=<variable value>
```

where:

- <Web Base server IP> is the IP address of the Web Base server.
- <first name> is the first name of the agent.
<last name> is the last name of the agent.
- The ani and dnis parameters and their associated <variable value> are optional.

Example: ani=18005555555 or dnis=4023

To get the call status for a specific agent:

Note: This example also applies if you used skilltarget_id or sender_id instead of username as the ID parameter. If you use all three of these ID parameters in the command, the search will return information only for the first ID parameter in the command.

Example: `http://<Web Base server IP>/recordingcontrols/rest/call_status?username=joe.smith`

If an active call was not answered, you will see the following response:

```
HTTP/1.1 200 OK
Date: Tue, 27 May 2014 15:28:23 GMT
Transfer-Encoding: chunked
{"contactId":"null","callActive":"true"}
```

A response usually includes the `contactId` and the `callActive` parameters. When `callActive` is true, Quality Management sees the call and plans to record it.

If the active call was answered, you will see the following response:

```
HTTP/1.1 200 OK
Date: Tue, 27 May 2014 15:28:23 GMT
Transfer-Encoding: chunked
{"contactId":"79","callActive":"true"}
```

If there is no active call, you will see the following response:

```
HTTP/1.1 200 OK
Date: Tue, 27 May 2014 15:46:41 GMT
Transfer-Encoding: chunked
{"contactId":"null","callActive":"false"}
```

To get the call status for a specific agent and DNIS:

Note: This example also applies if you used ANI or ANI and DNIS instead of DNIS. Or you used `skilltarget_id` or `sender_id` instead of `username`.

Example: `http://<Web Base server IP>/recordingcontrols/rest/call_status?username=joe.smith&ani=3002`

If there is an active call, you will see the following response:

```
HTTP/1.1 200 OK
Date: Tue, 27 May 2014 15:28:23 GMT
```

```
Transfer-Encoding: chunked  
{"contactId":"79","callActive":"true"}
```

If there is an active call and it is not being recorded, you will see the following response:

```
HTTP/1.1 200 OK  
Date: Tue, 27 May 2014 15:48:39 GMT  
Transfer-Encoding: chunked  
{"status":"User Joe is not configured for recording"}
```

Post-Call Survey API

The Post-Call Survey API is used by the client and the server. The Post-Call Survey API provides a means of importing customer surveys into Recording and Quality Management.

The survey application (that is, the system that performs the survey) is independent from Recording and Quality Management. You can use any type of interactive survey method (such as, IVR, email, or SMS). There are two CSV files: a Form CSV file and a Results CSV file. The survey application must write the results of the survey to a Results CSV file and the actual survey to the Form CSV file.

The interactive survey method greets the inbound caller and, through a voice script, determines whether or not the caller should be offered the option to take a post-call survey. If the caller agrees to be a survey candidate, the interactive survey method script generates a unique identifier (Survey ID) and determines the survey form (Form ID) to present at the conclusion of the call.

Recording and Quality Management assigns the associated call ID for a call to the post-call survey. The post call ID uses the associated call ID as the survey ID so a call can be matched with its post call survey.

Recording and Quality Management retrieves the watched folder. The default folder is located at: C:\Program Files\Common Files\QM\surveys. You can change the default location in the fileobserver.properties file located at C:\Program Files\Cisco\WFO_QM\config on the Operations server.

Recording and Quality Management retrieves the CSV files watched folder when it is updated and imports the survey results using the Survey ID as the common key. The Media Player displays the survey results in the Post-Call Survey tab for a selected call.

Post-Call Survey API Requirements

The survey application is independent from Recording and Quality Management. You can use any type of interactive survey method (such as, IVR, email, or SMS). The survey must:

- Contain a Survey ID
- Use the CSV format

CSV Format

The file name convention for the CSV files are as follows:

CSV file	CSV Format
Form	File name: Form_<FormID>.csv CSV format: <Form Name/Description>, <Status=editable/active><Date Created (yyyy-dd-MM)>, <Total Points> <Question #>, <Question Type>, <Question>, <Question Weight>

CSV file	CSV Format
Results	<p>File name: Results_<yyyyMMdd>_<HHMM>_<Uniquelidentifier>.csv</p> <p>where:</p> <ul style="list-style-type: none"> ■ <Uniquelidentifier> is an ID that is unique to each CSV file. The following example uses the UCID. You could specify a counter or any value that ensure the CSV file has a unique name. ■ <yyyyMMdd> is the year, month, and date. ■ <HHMM> is the time in hours and minutes. <div style="border: 1px solid black; background-color: #e6f2e6; padding: 5px; margin: 10px 0;"> <p>Note: A single digit for month, day, or hour must contain a leading zero to be considered valid.</p> </div> <div style="border: 1px solid black; background-color: #e6f2e6; padding: 5px; margin: 10px 0;"> <p>Example: Results_20140817_1938_16858473654321.csv</p> </div> <p>CSV format</p> <p><UCID (associatedCallId)>,<Form ID>,<survey Total Score>,<Question #>,<Result>,<Received score/weight></p> <p><UCID (associatedCallId)>,<Form ID>,<survey Total Score>,<Question #>,<Result>,<Received score/weight></p> <p><UCID (associatedCallId)>,<Form ID>,<survey Total Score>,<Question #>,<Result>,<Received score/weight></p>

Post-Call Survey Commands

Messages are sent to the Post-Call Survey API on the Operations server. By default, Post-Survey API listens on port 80 or 443 for incoming survey commands. The messages require the following information:

- Protocol: HTTP
- IP Address: <Web Base server>
- Port: 80 if you are not using HTTPS and 443 if you are using HTTPS

Command Syntax

The Post-Call Survey API supports the following HTTP GET method:

```
http://<Web Base server  
IP>/api/rest/recording/<command>/<associatedCallID>
```

where

- <Web Base server IP> is the IP address of the Web Base server.
- <associatedCallID> is the associated Call ID.

To get the a post-call survey for a specific associated call ID:

Example: `http://10.191.205.232/api-rest-recording/surveyForm/291317843898779`

To get all active post-call surveys:

Example: `http://10.191.205.232/api-rest-recording/surveyForm?status=active`

Note: This example returns active forms. Only active forms appear in the Recent Surveys widget in Unified Workforce Optimization.

To search for a specific survey score:

Example: `http://10.191.205.232/api-rest-recording//contact?expand=metadata&expand=eventCalculations&reason=recorded&range=date_range_in_the_past_year&survScore=75~less&dojo.preventCache=1406208488884`

Contact Basic Search API

The Contact Basic Search API returns details about an in progress or most recently completed call. The returned information consists of the most recent contact that matches the parameters of the search. The most recent contact might currently be in progress.

Contact Basic Search Commands

Messages are sent to the Contact Basic Search API on the Operations server. By default, Contact Basic Search API listens on port 80 or 443 for incoming search commands. The messages require the following information:

- Protocol: HTTP
- IP Address: <Web Base server>
- Port: 80 if you are not using HTTPS and 443 if you are using HTTPS

Command Syntax

The Contact Basic Search API supports the following HTTP GET method:

```
http://<Web Base server  
IP>/api/rest/recording/contactbasicsearch?<search query  
parameters>
```

where:

- <Web Base server IP> is the IP address of the Web Base server.
- <search query parameters> is the search criteria used to filter results. You can combine query parameters in a search. The supported query parameters are as follows:
 - ani
 - dnis
 - firstName
 - lastName

- username—for users in a domain, you can search with domain\user or username.
- displayID—the format of the display ID is as follows:

User Type	Format
QM User	0.<database Person ID>
Unified CCX	1.<loginID>

Note: The query parameters are case sensitive.

To search for a specific user:

Example:

```
http://10.192.247.197:80/api/rest/recording/contactbasicsearch?displayId=1.6000
```

Response:

```
{
  "id" : 2,
  "assocCallId" : "00001007771411573215",
  "recordingUrl" :
  "http://10.192.247.197:80/cwfo/apps/Recordings.html?loadContact=2",
  "isComplete" : false,
  "agent" : {
    "$ref" : "/api/rest/recording/person/3",
    "displayId" : "2.6000",
    "lastName" : "test agent 1",
    "firstName" : "",
    "username" : "a1"
  }
}
```

Customer Relationship Management Integration

Cisco allows you to configure your customer relationship management (CRM) system to send URL commands to the following Quality Management versions through the Recording Controls API.

- Quality Management 10.5(1) SR2 or later and 11.0(1) or later

The CRM system must support custom buttons, links, and workflow actions.

Before you configure your CRM system, you need to determine which URL commands you want to send to Quality Management. See the *API Programmer Guide* for information on the commands and syntax for the Recording Controls API.

You also need to determine the type of metadata that you want to use. Quality Management Administrator allows you to create up to 10 user-defined metadata fields. The metadata can be sent from your CRM system to Quality Management.

Best Practices

The following steps are recommended when integrating Quality Management with a CRM system:

1. From Quality Management Administrator, define your required metadata. See the *Administrator Guide* for instructions.

Example: Create a metadata field for the Salesforce number (SF#).



2. From your CRM system, build an object.

This object can be related to another object. You can use the object relationship to connect a recorded call object to an object in your CRM database. This specific object will contain links to each call recording in Quality Management.

Example: You can create an object for recorded calls. The following figure shows an object called Recorded Calls.



3. Create a URL request for the object that sends metadata to Quality Management. This metadata uniquely identifies the recorded call so that you can locate it later.

Send metadata request URL format example:

```
http://<QM base server>/recordingcontrols/rest/metadata?userdomain=<domain name>&Username=<username>&<metadata field>="<value>"
```

where:

- <QM base server> is the IP address or hostname of the Quality Management base server
- <domain name> is the domain name

- <username> is the agent's username
- <metadata field> is the name of the metadata field configured in Quality Management Administrator
- <value> is the data associated with the metadata field

Send metadata request URL example:

```
http://10.194.225.163/recordingcontrols/rest/metadata?userdomain=acme&Username=john.doe&SF#="1532"
```

4. In the CRM database, define a workflow action, button, or link that will send the URL request to the Cisco Recording Controls REST API.
5. From your CRM system, create a URL that will search for data sent to Quality Management with the previous URL.

Search URL format example:

```
http://<QM base server>/cwfo/app/Recordings.html?userLang=<en>&userCountry=<NULL>&urlSearch=true&userdomain=<domain name>&metadata=<metadata field>~<value>
```

Note: The search URL must include the userLang and userCountry fields and the metadata field name.

Search URL example:

```
http://10.194.225.163/cwfo/app/Recordings.html?userLang=<en>&userCountry=<NULL>&urlSearch=true&userdomain=acme&metadata=SF#~1532
```

This URL automates the creation of a link to the recording in Quality Management.