Cisco Unified Workforce Optimization

Quality Management API and Database Schema Reference Guide Version 11.5

First Published: July 28, 2016

Last Updated: August 9, 2018

Cisco Systems, Inc. www.cisco.com

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.

Quality Management API and Database Schema Reference Guide

Contents

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 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.

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 are 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 is:

```
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
```

HTTP Request CSRF Token

In order to enforce CSRF security, an attached CSRF token is required for all non-`GET` requests made to the surrogate application (viz. the base URI, the "/api" URI, and any SSRS proxy URIs) that have an active session token. If the CSRF token is not attached or is not valid, the application will provide a 401 response and the request will not be completed.

There is a one-to-one correspondence between session tokens and CSRF tokens. The CSRF and session token pair is passed to the client in a single response via the `CSRFTOKEN` and `JSESSIONID` cookies respectively. Any response that contains a session token will include a CSRF token as well. Clients must maintain the value for the `CSRFTOKEN` cookie and attach this value to all subsequent requests.

The CSRF token can be attached to requests in two ways:

1. The `X-CSRF-Token` header
2. The `csrfToken` request body parameter

The `X-CSRF-Token` header is the preferred method when possible, such as when using AJAX or an HTTP client API. The csrfToken request body parameter is generally required for HTML forms.

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, for 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.

To start 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.

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.

Parameter	Description
agent	A person's ID. The accepted value is integer ID. This parameter is optional.
ani	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. This parameter can appear zero or more times in a single query. **Example:** `contact?ani=1234&ani=4567` 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 ani=1234 OR ani=4567). This parameter is optional.
assocCallId	The Associated Call ID that ties together contacts based on a customer experience. This parameter is optional.
beginTime	Return only records that start on or after the specified date. When specifying a date, choose one of the following date formats: - `YYYY-MM-DD HH:MM:SS` - `YYYY-MM-DD` This parameter is optional.

Parameter	Description
canEvaluate	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. Accepted Values: "true” if this contact should be returned if the contact can be evaluate by the logged in person. This parameter is optional. If you do not include this parameter, the query does not filter the contacts.
dnis	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 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.
endTime	Return only records that start before the specified date. When specifying a date, choose one of the following date formats: ■ `YYYY-MM-DD HH:MM:SS` ■ `YYYY-MM-DD` This parameter is optional.
exclude	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: ■ 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. This parameter is optional.

Parameter	Description
expand	Returns all data associated with the specified value instead of a URI. The accepted values are: ■ 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. This parameter is optional.
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	Return only contacts associated with a recording. The accepted values are: ■ true ■ false This parameter is optional. If you do not include this parameter, the query does not filter contacts by recordings.
hr	Whether the contact evaluation has been marked for human resources (hr). The accepted Boolean values are: ■ true ■ false ■ 1 (true) ■ 0 (false) This parameter is optional.

Parameter	Description
inProgress	Whether the contact evaluation is in progress. The accepted Boolean values are: ■ true ■ false ■ 1 (true) ■ 0 (false) This parameter is optional.
lastName	The agent's last name. The accepted values is string with any number of asterisk (*) or question mark (?) wildcards. This parameter is optional. 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.
line	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. 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.
metadata	The metadata field. The accepted values are: ■ <key> ■ <key>~<value> 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. This parameter is optional.

Parameter	Description
needsApproval	Whether the contact evaluation needs approval. The accepted Boolean values are: ■ true ■ false ■ 1 (true) ■ 0 (false) This parameter is optional.
number	Any number used in the contact (ANI, DNIS, or Line). The accepted value is 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. This parameter is optional.
scored	Whether the contact has been scored. The accepted Boolean values are: ■ true ■ false ■ 1 (true) ■ 0 (false) This parameter is optional.
silenceDuration	Return only contacts where the duration (in milliseconds) of recorded silence is equal to or greater than the value specified. **Example:** silenceDuration=5000 returns contacts where there are one or more instances of recorded silence equal to or greater than 5 seconds. This parameter is optional.

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: ▪ true ▪ false ▪ 1 (true) ▪ 0 (false) This parameter is optional.
talkOverDuration	Return only contacts where the duration (in milliseconds) 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	Whether the contact evaluation has been marked for training. The accepted Boolean values are: ■ true ■ false ■ 1 (true) ■ 0 (false) This parameter is optional.
type	The type of contact. The type parameter filters contacts based on upload states. The accepted values are: ■ quality ■ archive 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.

Examples

Combination search:

The following request returns a list of all contacts recorded for quality purposes on or after 01/01/2015 (GMT).

```
GET ~/api/rest/recording/contact?beginTime=2015-01-
01&type=quality
```

Searching for a metadata key:

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

```
GET ~/api/rest/recording/contact?beginTime=2015-01-
01&metadata=phone
```

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

```
GET ~/api/rest/recording/contact?beginTime=2015-01-
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=2015-01-
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 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"
   }
 }
```

To support a single contact:

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 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 CAD, see the *Quality Management Integration Guide for CAD and Finesse*.

Hot Desking

Hot desking is a situation where one desk is shared by 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 User 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—use the userdomain and username as the agent identifier 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 buttons that are available to an agent or knowledge work are configured from Role Permissions window in Quality Management Administrator. See "Recording Control Buttons" in the *Administrator Guide* for more information.
- 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 more information about the Recording Controls Browser application, see the *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	Records a call and uploads the call to the Quality Management server at the end of the day. In the Recording Controls API, the <command> is record. The Record Tag command behaves as follows: ■ 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 *Administrator Guide* for more information. Quality Management stores agent-tagged calls with the Agent Tagged reason code, and saves them for the retention time configured in Quality Management Administrator. Agent Recording: ■ 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.

Command	Function
	■ 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 the agent, based on the call start times, and tags the first call when you invoke the Record Tag command. Gateway/MediaSense Recording and Agent Recording: ■ 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	Temporarily halts the recording of: ■ 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. `Screen recording paused` 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. 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. When using the Pause command, note the following: ■ Agent Recording: • 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. Issue the Resume command when you want to start recording after a pause.

Command	Function
Resume	Resumes recording after you issued a Pause command to stop the recording. Agent Recording: ■ The Resume command affects voice and screen recording. ■ If the call is not currently paused, the Resume command has no effect. ■ The Resume command is valid for active calls only. ■ If you do not use the Resume command, the point at which you paused the recording is the end of the audio recording. ■ A Resume command does not appear as a mutual silence event or talkover event during post-call processing.

Command	Function
Restart	Restarts or starts the recording of a call. **Note:** The Restart command is not supported with Gateway Recording and will be removed in a future release. In the Recording Controls API, the <command> is restart. Agent Recording: ■ 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. ■ If Quality Management is not currently recording an active call, the Restart command starts audio and screen recording. ■ The Restart command is valid for active calls only. ■ 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. Gateway/MediaSense Recording does not support the Restart command. 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).

Command	Function
Delete	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. In the Recording Controls API, the <command> is delete. ■ 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 Quality Management. ■ 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	Sends a login request that associates an agent with the specific extension for hot desking. 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. In the Recording Controls API, the <command> is login. You must include the unique extension of the phone that the agent is logging into. This command is not supported if you are using Gateway/MediaSense Recording.

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

Command	Function
Metadata	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. In the Recording Controls API, the <command> is metadata. You must include at least one key/value pair (<key>=<value> or <key>:<value>). ■ 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. 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. **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. Successive calls to the Metadata command using the same key

Command	Function
	name update the existing metadata for that call. Specifying an empty value for a key removes that metadata field association for the call. Valid formats for metadata are as follows. ■ 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. **Example:** & or = All other alphanumeric characters are valid. You can find the decimal point in the * key menu and the dash in the zero key menu on your phone.

Command	Function
Start Segment	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. In the Recording Controls API, the <command> is start. Agent Recording: ■ If Quality Management is not currently recording an active call, the Start Segment command starts audio and screen recording. ■ If Quality Management is currently recording an active call, the Start Segment command has no effect. ■ If the active call ends before the recording is stopped by the agent, the recording is saved according to workflow criteria. ■ The Start Segment command does not override the workflow. ■ The Start Segment command does override the exclusion list in the Exclusion List window. **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. Gateway/MediaSense Recording does not support the Start Segment command.

Command	Function
Stop Segment	Stops the audio and screen recording of an active call. The recording is then saved according to workflow criteria as a new contact. In the Recording Controls API, the <command> is stop. ■ Agent Recording supports the Stop Segment command only during active calls. ■ Gateway/MediaSense Recording does not support the Stop Segment command. 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.

Command	Function
Start Screen	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. Voice contact recordings and screen only contact recordings can be bracketed or interleaved. **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. In the Recording Controls API, the <command> is start_screen. Agent Recording: - 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. Gateway/MediaSense Recording does not support the Start Screen command. 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

Command	Function
	behavior. ■ 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. **Note:** The Restart command is not supported with Gateway Recording and will be removed in a future release. ■ 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 All other commands have no affect on the current screen only recording. Issue the Stop Screen command to stop screen only recording.

Command	Function
Stop Screen	Stops screen recording. In the Recording Controls API, the <command> is stop_screen. Agent Recording: ■ 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. Gateway/MediaSense Recording does not support the Stop Screen command.
Config	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.

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.

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.
Restart	Recording restarts. Sets Reason Code to TAGGED.	Recording starts. Sets Reason Code to TAGGED.

Command	Currently Recording	Not Currently Recording
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.

HTTP Status Codes

The Recording Controls API returns HTTP status codes that indicate whether a command was successfully executed. These status codes are as follows.

Status Code	Description
200 OK	The command was executed successfully.
400 Bad Request	The command did not execute due to a configuration issue or invalid parameters.
401 Unauthorized	The command did not execute due to permission issues.
500 Internal Server Error	The command did not run due to an internal server error.

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 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 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.

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=true	Yes	No
Segment and Save	Yes	No
Segment and Delete	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.

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.log
log4j.appender.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.dbg
log4j.appender.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.title=Cisco Recording Controls
```

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.

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
```

Upgrading Quality Management

If you upgrade your version of Quality Management, any changes to the recordingcontrols.properties files are lost in the upgrade. Back up the recordingcontrols.properties file if you want to save your changes to this file.

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.

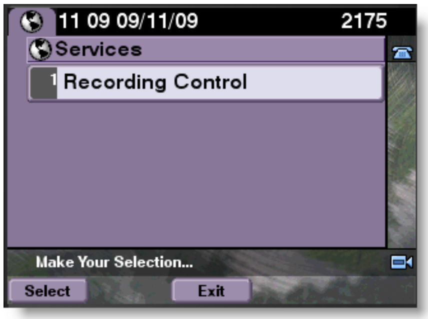
   The Cisco IP Phone Service can only be used by Quality Management agents who are using the Network Recording service.

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.



11 09 09/11/09 2175

Services

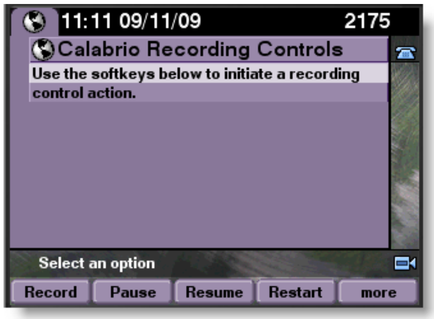
1 Recording Control

Make Your Selection...

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.



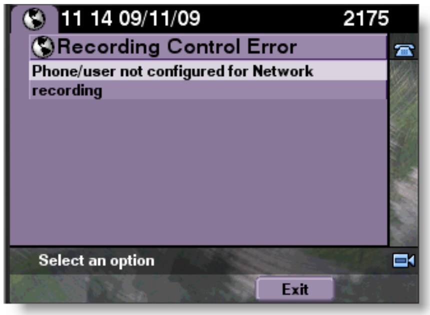
11:11 09/11/09 2175

Calabrio Recording Controls

Use the softkeys below to initiate a recording control action.

Select an option

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



11 14 09/11/09 2175

Recording Control Error

Phone/user not configured for Network recording

Select an option

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 Ciscoo 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 Integration 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 a call identifier for a call to the post-call survey. The call identifier acts as the survey ID so a call can be matched with its post call survey. The possible call identifiers are as follows:

- Associated Call ID—the survey connects to all contacts with this call identifier.
- Contact ID—the survey connects to a single contact with this call identifier. This call identifier is located in the results column of the first question in the results file.
- ICM Call ID—the survey connects to a specific contact with this call identifier.

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\Ciscoo\WFO_QM\config on the Operations server.

Recording and Quality Management retrieves the CSV files from the 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 where <FormID> is a number. CSV format: <Form Name/Description>,<Status=editable/active><Date Created (yyyy-MM-dd)>,<Total Points> <Question #>,<Question Type>,<Question>,<Question Responses and Weights (option - text for response - weight)> where <Form Name/Description> must be unique.

CSV file	CSV Format
Results	File name: Results_<yyyyMMdd>_<HHmm>_<UniqueIdentifier>.csv where: ■ <UniqueIdentifier> 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. **Note:** A single digit for month, day, or hour must contain a leading zero to be considered valid. **Example:** Results_20140817_1938_16858473654321.csv CSV format <UniqueIdentifier>,<Form ID>,<survey Total Score>,1,<UCID (CallIdentifier)>,<Received score/weight> <UniqueIdentifier>,<Form ID>,<survey Total Score>,2,<AGENTID (Agent ID)>,<Received score/weight <UniqueIdentifier>,<Form ID>,<survey Total Score>,<Question #>,<Result>,<Received score/weight> <UniqueIdentifier>,<Form ID>,<survey Total Score>,<Question #>,<Result>,<Received score/weight>

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>/<callIdentifier.
```

where

- <Web Base server IP> is the IP address of the Web Base server.
- <callIdentifier> is the Associated Call ID, Contact ID, or ICM Call ID. The <callIdentifier> for a survey is defined in Recordings > Survey Form Administration in Quality Management Administrator. See "Survey Form Administration" in the *Administrator Guide* for more information.

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"
     }
}
```

Import API

The Import API will be responsible for creating Ccr, Media, and Media File table entries from POST requests submitted by a custom written customer application or other HTTP POST methods. The Import API will return the relevant contact ID created along with the Media objects. Media objects will respond with the corresponding URL of where each individual media file should be sent for Media Upload which includes a hashed signature. The customer application will then be responsible for sending a POST for each file to the appropriate URL path. The existing Cisco Media service will be leveraged to manage uploading of the media files to the proper SAN/NAS location and setting the appropriate upload state of the Media Files and Ccr table entries.

General process flow:

In order to complete an upload there will be a series of API calls.

1. Files must be unencrypted (.wav) for audio or unencrypted (webM) for video.
2. 3rd party uses a REST API call to create an authorized session.
3. 3rd party uses a REST API call to identify the contact being created including metadata which will be inserted into the Cisco database.
4. Cisco Media service responds with Media URL location(s) on where contacts created should be stored.
5. 3rd party uses a REST API to POST corresponding media to URLs returned in Contact Creation (step 4).

Import API High Level Diagram

.

JSON Object Request Examples

Note: As part of the Contact Creation (metadata), the Import API will need to know who made the call. This login name can change slightly depending on the authentication method being used. If Quality Management is using Active Directory for authentication then Login.Name and Domain need to be sent in the request. If using Local QM Authentication then Login.Name is sufficient but DisplayID can also be used for this authentication environment.

- Name—The Login name of the person configured in Recording and Quality Management.
- Domain—If Active Directory Authentication Is used this is the Domain of the person configured in Quality Management.

- DisplayID—Alternate unique string for identifying the person. DisplayID can be found in the following locations:
  - Quality Management database <> Person Table under DisplayID
  - Quality Management Administrator > Personnel > User Administration > User ID

Example 1: When Cisco QM System is Integrated to Active Directory for Authentication

```
{"Person":
{"Name":"Joe.Smith","Domain":"someDomain"},"AssocCallId":"291602527108185
","CallId":"17513561","CalledAddress":"9996","CallingAddress":"9998","Cli
entTimeZone":"Pacific Standard Time","Direction":1,"Files":
[{"fileName":"somename1","FileType":"Audio"},
{"fileName":"somename2","FileType":"Screen","Coordinates":"0,0","DisplayN
ame":"DISPLAY1","Resolution":"1280x800"},
{"fileName":"somename3","FileType":"Screen",
"Coordinates":"0,20","DisplayName":"DISPLAY2","Resolution":"1280x800"}],"
Line":"9998","SiteName":"SomeSite","StartTime":1426194888,"TelephoneGroup
":"SomeTelephonyGroup","Type":"RecordingTypeName"}
```

Example 2: When Cisco QM system is only using Local QM Authentication

```
{"Person":
{"DisplayId":"SomeId"},"AssocCallId":"291602527108185","CallId":"17513561
","CalledAddress":"9996","CallingAddress":"9998","ClientTimeZone":"Pacifi
c Standard Time","Direction":1,"Files":
[{"fileName":"somename1","FileType":"Audio"},
{"fileName":"somename2","FileType":"Screen","Coordinates":"0,0","DisplayN
ame":"DISPLAY1","Resolution":"1280x800"},
{"fileName":"somename3","FileType":"Screen",
"Coordinates":"0,20","DisplayName":"DISPLAY2","Resolution":"1280x800"}],"
Line":"9998","SiteName":"SomeSite","StartTime":1426194888,"TelephoneGroup
":"SomeTelephonyGroup","Type":"RecordingTypeName"}
```

Detailed Import Process

Step 1: Authorization

Request

POST to http://<base server>/api/rest/authorize

Body:

```
[
  {
    "id":"recording",
    "userId": "importUser",
    "password": "PCS997#62",
    "data": {
    "qm.service":true
    }
  }
]
```

Response

Extract and save cookie header.

```
Set-Cookie: JSESSIONID=12da1y8o4k9yt1iq0zzb0tgzbw;Path=/
```

Body:

```
[
  {
    "id":"recording",
    "status":200,
    "userName":{
      "first":"ImportUser",
      "last":"System"
    },
    "lang":"en",
    "country":""
  }
]
```

Note: Authorization will always return the Set-Cookie header, even in the case of an error. You must check that the response code or the status field in the response body was 200.

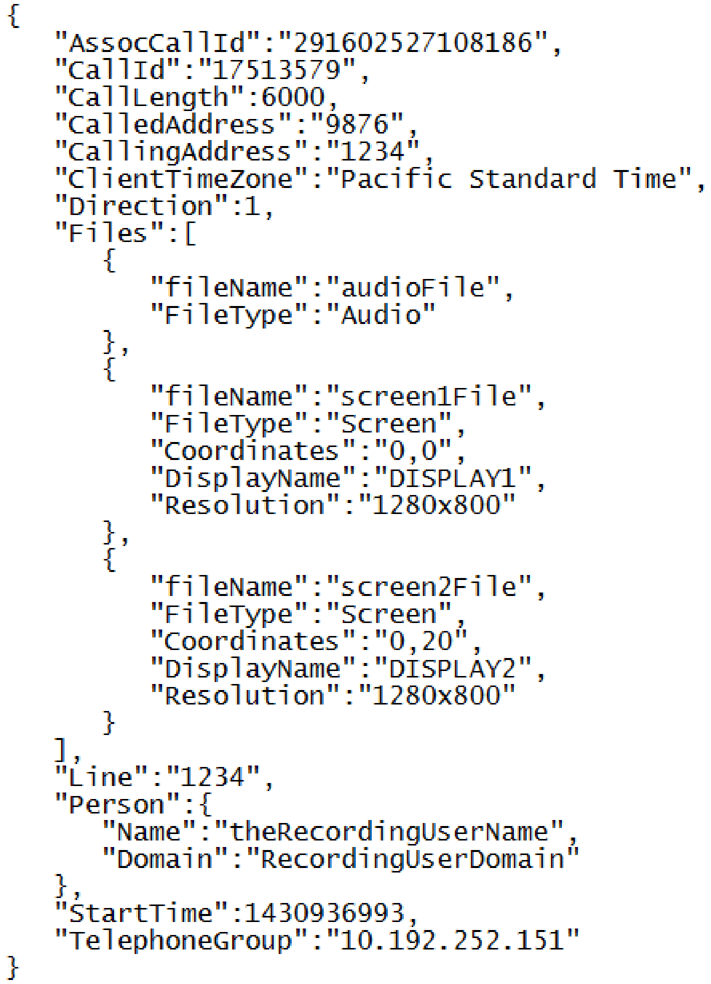
Step 2: Contact Creation

Request

```
POST to http://<server ip>/api/rest/recording/contact/upload
```

Include Set-Cookie header from Authorization.

Body:



```
{
   "AssocCallId":"291602527108186",
   "CallId":"17513579",
   "CallLength":6000,
   "CalledAddress":"9876",
   "CallingAddress":"1234",
   "ClientTimeZone":"Pacific Standard Time",
   "Direction":1,
   "Files":[
      {
         "fileName":"audioFile",
         "FileType":"Audio"
      },
      {
         "fileName":"screen1File",
         "FileType":"Screen",
         "Coordinates":"0,0",
         "DisplayName":"DISPLAY1",
         "Resolution":"1280x800"
      },
      {
         "fileName":"screen2File",
         "FileType":"Screen",
         "Coordinates":"0,20",
         "DisplayName":"DISPLAY2",
         "Resolution":"1280x800"
      }
   ],
   "Line":"1234",
   "Person":{
      "Name":"theRecordingUserName",
      "Domain":"RecordingUserDomain"
   },
   "StartTime":1430936993,
   "TelephoneGroup":"10.192.252.151"
}
```

Required Fields

Field	Description
CallLength	Duration of call in milliseconds.
TelephoneGroup	The name of the telephony group.

Field	Description
Files	Media associated with the contact (at least one). The required subfields are as follows: ■ fileName: Names of files in the database and on the storage location are auto-generated. The file names are used to associate files with upload URLs. ■ fileType: The type of file. Your options are Audio or Screen. The optional subfields applies to Screen only, and are null if Screen is absent: ■ Coordinates: Pairs of relative screen locations in the following format: x,y ■ DisplayName: The name of the display screen. ■ Resolution: The size of the screen in pixels in the following format: width x height
Person	The person associated with the contact.
Name	The username of the recording user. This field is required for LDAP Authentication and QM Authentication.
Domain	The domain name of the recording user. This field is only required for LDAP Authentication.

Field	Description
DisplayId	The person associated with the contact. The required subfields are as follows: ■ Name: The username of the recording user. This field is required for LDAP Authentication and QM Authentication. ■ Domain: The username of the recording user. This field is required for LDAP Authentication and QM Authentication. You can optionally use the following subfield: ■ DisplayId: The user ID. This is an acceptable alternative for LDAP Authentication and QM Authentication. The format is as follows: realm.skillTargetId

Optional Fields

Optional fields are null if absent, unless specified otherwise.

StartTime	The beginning of the call as a unix/epoch timestamp (seconds since 1/1/1970). Current server timezone will be used if absent.
ClientTimeZone	The timezone on the client (for example, Pacific Standard Time). Current server timezone will be used if absent.
Direction	The direction of the call. 1 for inbound calls, 0 for outbound calls. 1(inbound) will be used if the value is absent.
AssocCallId	Maps to Ccr.associatedCallId.
CallId	Maps to Ccr.icmCallId.
CalledAddress	Maps to Ccr.dnis.
CallingAddress	Maps to Ccr.ani.
Line	Maps to Ccr.line.

Response

Body:

```
{
  "ContactId":88,
  "Files":[
    {
      "fileName":"audioFile",
      "UploadPath":"http://<siteip>/media/upload/88?timestamp=1431010756&
signature=BD00EC5AA30AA56761BED14456556596CCFB1CC0"
    },
    {
      "fileName":"screen1File",
      "UploadPath":"http://<siteip>/media/upload/89?timestamp=1431010756&
signature=AF858B426A913A4E10969D805C7CCDA1D9835CEB"
    },
    {
      "fileName":"screen2File",
      "UploadPath":"http://<siteip>/media/upload/90?timestamp=1431010756&
signature=2FE41B5D14640539913F4B8F13BA76A675E9BA38"
    }
  ]
}
```

Step 3: File Upload

Request

POST to URLs returned from Contact Creation (that is, the UploadPath).

Headers:

Content-Type: audio/x-wav or video/webm

Content-Length: (This should be automatically included, but service will verify that it is correct.)

Body: corresponding media file

Response

body:

```
{
   "result":200,
   "message":"success"
}
```

General Notes

JSON in examples has been reformatted for reading clarity. Services will accept this formatting but do not require it. All responses will be returned unformatted.

Defining the Quality Reason on Imported Calls

By default, all contacts uploaded via the /api/rest/recording/contact/upload API have a archive reason and quality reason of `15:API Imported`. If you want to have an alternate reason code, use the optional JSON top-level attributes `archiveReason` and `qualityReason` . You must enter a valid Reason Code as the Reason ID, as defined in RecordingReason table. If you enter an invalid ID or no value, the value will default to `15:API Imported`.

Example:

```
"archiveReason":"14",
"qualityReason":"0",
```

Errors

Authentication

Errors will be in the format of a JSON message. For example:

```
{"errorMessage":"Exception parsing authentication request."}
```

Authentication might also return a 401 response code with no body.

Contact Creation

Errors will be in the format of a JSON message. For example:

{"errorMessage":"No person found for data: {\"Name\":\"Steve\"}"}

File Upload

Errors will be a HTML page. For example:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"/>
<title>Error 500 Request validation failed: All requests must have one
signature parameter.</title>
</head>
<body><h2>HTTP ERROR 500</h2>
<p>Problem accessing /media/upload/92. Reason:
<pre> Request validation failed: All requests must have one signature
parameter.</pre></p><h3>Caused by:</h3><pre>java.io.IOException: Request
validation failed: All requests must have one signature parameter.
        at com.Cisco.qm.media.MediaApiServlet.service
```

```
(MediaApiServlet.java:169)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:848)
```

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, 11.0(1) or later, and 11.5(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 *Recording Controls API*for information on the commands and syntax to use.

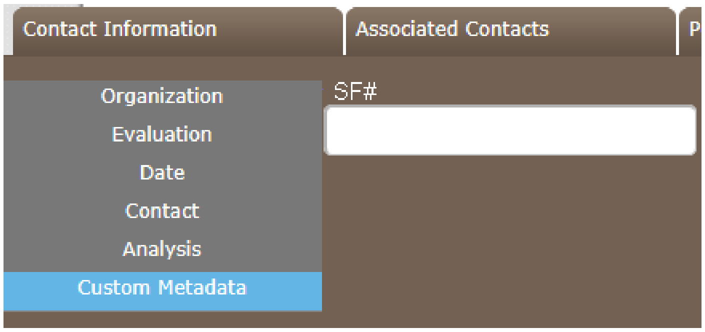
You also need to determine the type of metadata that you want to use. Quality Management Administrator allows you to create up to 10 user-defined metadata fields. The metadata can be sent from your CRM system to Quality Management.

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.

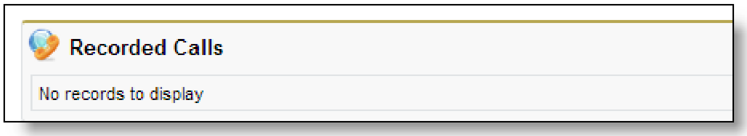
   Example: Create a metadata field for the Salesforce number (SF#).



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.


Recorded Calls

No records to display

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 located it later.

   Send metadata request URL format example:

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

   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"~equals

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.

Database Schema

This sections describes how data is organized in the Cisco Quality Management database, defines the fields in each table, and describes changes to the database schema.

General Database Features

The following topics provides an overview of some of the basic database features.

Dates in the Database

Quality Management stores most timestamps in GMT. You can convert GMT timestamps to local timestamps for display or to compare with local time ranges by using the time zone offset (TzOffset) that always accompanies the GMT time. Activated and deactivated dates, however, are stored in GMT without the offset for local time because these do not need to be converted to local time. The Ccr table also stores the local time already calculated because it is used frequently to compare with local time ranges, such as in Quality Management Reporting.

Two terms are used here and as database column name suffixes:

- TzFK—Time Zone Foreign Key; a foreign key into the Timezone table. A value of zero indicates that the time zone is currently unknown. Any time zones that remain unknown will not be updated and these timestamps will be displayed in GMT.
- TzOffset—Time Zone Offset from GMT; a value representing a positive or negative offset in increments of 15 minutes from GMT (UTC). Adding (TzOffset – 128) × 15 minutes to the corresponding (GMT) timestamp will always provide the local time. The TINYINT type, used in Microsoft SQL Server to store TzOffset values, is an unsigned 8-bit value. A value of 128 indicates “no offset”, and this value is always used when the TzFK is zero. The possible TzOffset values are shown in the following table.

Hours	TzOffset	Signed
-23:45	33	33
-23:00	36	36
-22:00	40	40
-21:00	44	44
-20:00	48	48
-19:00	52	52
-18:00	56	56

Hours	TzOffset	Signed
-17:00	60	60
-16:00	64	64
-15:00	68	68
-14:00	72	72
-13:00	76	76
-12:00	80	80
-11:00	84	84
-10:00	88	88
-09:00	92	92
-08:00	96	96
-07:00	100	100
-06:00	104	104
-05:00	108	108
-04:00	112	112
-03:00	116	116
-02:00	120	120
-01:00	124	124
-00:15	127	128
00:00	128	132
00:15	127	136
+01:00	132	140
+02:00	136	144
+03:00	140	148
+04:00	144	152
+05:00	148	156
+06:00	152	160

Hours	TzOffset	Signed
+07:00	156	164
+08:00	160	168
+09:00	164	172
+10:00	168	176
+11:00	172	180
+12:00	176	184
+13:00	180	188
+14:00	184	192
+15:00	188	196
+16:00	192	200
+17:00	196	204
+18:00	200	208
+19:00	204	212
+20:00	208	216
+21:00	212	220
+22:00	216	224
+23:00	220	228
+23:45	223	232

Quarter-hour values can be derived from the previous table by interpolation.

Example: TzOffzet value of 113 represents -0415 hours.

TzOffset values in the database are unsigned values ranging from 0 to 255. Java uses signed values for its 8-bit byte values, so those are also listed for reference.

Activation and Deactivation Dates in the Database

The Quality Management database stores both current and historical data.

Current data query example: “What teams are currently in this group?”

Historical data query example: “What teams were in this group one year ago?”

Historical data relies on activated and deactivated timestamps to determine the time window during which a record is valid. A special date value is used to bound the upper range of valid dates. The Activated column will always have a real date. The Deactivated column will have an upper bound of 2999-12-31 for records that are active.

Whenever a new row is added to a Quality Management table containing an activated column, that column is set to indicate the date and time when the row was added. This represents the first time at which the Quality Management has awareness of the record—prior to this time it is as if the record was not in the system.

Action	Activated Column is…	Deactivated Column is…
Activate a record (add a row)	Set to current date/time	Set to 2999-12-31
Deactivate a record	Never changed	Set to current date/time

Whenever a new row is added to a Quality Management table containing a deactivated column, that column is set to the 2999-12-31 value. This indicates that the record is active (as in “not yet deactivated”).

Deactivation means setting the deactivated column to a timestamp value less than 2999-12-31 .

Example: When an agent is removed from a team, Quality Management remembers this by deactivating (not deleting) the appropriate AgentTeam record.

In this example, Quality Management uses the time and date when the agent was removed from the team. The activated and deactivated columns for this row now specify the time window during which this agent was part of that team. To maintain data historically, records are deactivated, not deleted. The DB Cleaner service is the only application which deletes database records containing activated and deactivated columns.

In most cases, there is no such thing as reactivation of a record in Quality Management. An activated timestamp is only written once (when the record is created). The deactivated timestamp is set to 2999-12-31 when a record is created, and it is only updated once (when the record is deactivated). A record is never reactivated by resetting the deactivated timestamp to 2999-12-31 , since doing so would erase the fact that there was a period of time when the record was inactive. Instead, a whole new record would be created with the reactivation date being stored in the activated column. The exception to this occurs in the Person table. If a person is deactivated in the Quality Management database, the deactivated timestamp is set to the current time as in any other table. However, if the person is reactivated with the same realm.skillTargetId, Quality Management will reactivate the Person record by setting the deactivation timestamp to the 2999-12-31 value.

When checking if a record was active at a specified date and time, the activated conditional should be inclusive and the deactivated conditional should be exclusive.

Example: `Table.activated >= "2006-07-22 12:34" and Table.deactivated < "2006-07-22 12:34"`

DisplayIds in the Database Schema

In Quality Management, every agent and team is provided with a displayID. Quality Management recognizes each agent and team by means of a Peripheral Gateway (PG) and Peripheral Number (PN), which are associated with each displayID. The database schema uses the PG.PN format to identify the peripheral gateway and peripheral number associated with an agent.

For Agent.displayId, the possible PG values are:

- 0—Quality Management agents and knowledge workers
- 1—Cisco Unified Contact Center Express (Unified CCX) agents

For Agent.displayId, the possible PN values are:

- A unique ID generated by Quality Management—Agents and knowledge workers in the Quality Management database
- Resource.resourceLoginID—Agents in the Unified CCX database

Example: 0.2003 indicates the person is a Quality Management agent or knowledge worker.

For more information about the Resource.resourceLoginID and Agent.PeripheralNumber, see the *Database Schema Guide for Cisco Unified CCX and Cisco Unified IP IVR* available at www.cisco.com.

For Team.displayId, the possible PG values are:

- 0—Quality Management teams
- 1—Unified CCX teams

For Team.displayId, the possible PN values are:

- A unique ID generated by Quality Management—Teams in the Quality Management database
- Team.teamID—Teams in the Unified CCX database

Example: 1.Team.teamID indicates the team is a Unified CCX team.

Functions in the Database Schema

The Quality Management database schema includes several function definitions. These are used to simplify the SQL for certain commonly used query clauses. The possible functions are:

- **AddTzOffset**—this function adds a time zone offset value to a specified SQL date and time value, returning an adjusted date and time.

  Example: `SELECT dbo.AddTzOffset(Ccr.startTime, Ccr.startTimeTzOffset)…`

  Use this function to convert a GMT timestamp to one representing the local time.

- **TzOffsetToGMT**

  This function returns a time zone name string in the format <time-zone-abbreviation> HH:MM, for example, GMT 12:54. It will always be exactly 9 characters.

  Example: `SELECT dbo.TzOffsetToGMT`(Ccr.startTimeTzOffset)…

  This function was used in QM 2.3 and later for reporting to construct time zone name strings using only SQL. In Java code, Date objects are constructed using the proper time zone.

- **millisToHMS**

  This function returns a string in the format H:MM:SS, converting a milliseconds value to hours/minutes/seconds format.

  Example: `SELECT dbo.millisToHMS(Ccr.TimeToDrop - Ccr.timeToAnswer) AS duration…`

  Note: The millisToHMS function is not supported in QM 2.6 or later. It is in the Ccr table for historical purposes only. QM 2.6 or later uses callDuration to perform the same function. The callDuration is calculated and sorted in the database.

  The hours value in the result string can be up to 3 digits long, and is preceded by a minus sign (—) if the input value is negative. Thus, the returned string is always between 7 and 10 characters in length.

Database Table Details

This section provides information about Quality Management database tables, their records, and their fields.

Each description provides the following information:

- Database Table Name—Name of the Quality Management database table
- Column Name—Name of the columns as it appears in the database table
- Description—Explanation of the field, including valid values where appropriate
- Storage—Information about the data in each field as follows:
    - Data type used for the field in the database
    - Whether the NULL value is valid for the field. “NULL” if the NULL value is valid or “NOT NULL” if the NULL value is not valid
    - “Primary Key” if the field is a primary key, or part of a primary key, in the database table

The following table summarizes the complete list of tables for Quality Management database schema.

Name	Direction
ACD	Contains records written for the ACD
ACDQueue	Contains records written for the ACD queue
ACDServer	Contains records written for the ACD server
ACDType	Contains records written for the ACD type
Agent	Contains records written for agents and knowledge workers configured in Quality Management Administrator
AgentTeam	Relationship table between agents and teams
Alert	Contains records written for alerts
AlertType	Contains records for the possible alert types
ApprovalType	Contains records for the approval type.
ArchiveAudit	Contains records written for every recording accessed in the Search and Play widget

Name	Direction
Ax1UCCMSchema	Contains records written for each version of Cisco Unified CM with an Administration XML (AXL) schema version in a telephony group
Ccr	Contains records written for every incoming, outgoing, or internal call
CcrType	Contains records written for the Ccr type
Cdr	Contains the records written for each associated contact
CdrCcr	Contains foreign keys to the Cdr and Ccr tables
DashRoleView	Contains records for the role associated with the Dashboard view
DashUserView	Contains records for the valid Dashboard views for each user
DashView	Contains records of the valid Dashboard views
DashWidget	Contains records of the valid Dashboard widgets
DashWidgetRole	Contains records of the valid roles for each widget
DashWidgetSetting	Contains records of the valid settings for each widget
DashWidgetView	Contains records of the valid views for each widget
DashWidgetViewSetting	Contains records of the valid settings for each Dashboard view
DbProperties	Contains various database properties and keys
Eval	Contains records written for every evaluation performed on a recording
EvalAlerts	Contains records written for evaluation alerts

Name	Direction
EvalComment	Contains records written for every comment made on an evaluation of a recording
EvalForm	Contains records written for every evaluation form created in Quality Management Administrator
EvalFormQuestion	Contains records written for every question added to an evaluation form in Quality Management Administrator
EvalFormQuestionOption	Contains records written for every new option added to a question
EvalFormQuestionOptionType	Contains records written for every new option type associated with a question
EvalFormQuestionTemplate	Contains records written for each new evaluation form question template
EvalFormQuestionTemplateOption	Contains records written for each new evaluation form question template option
EvalFormScoringType	Contains records written for each a new scoring type
EvalFormSection	Contains records written for every section added to an evaluation form in Quality Management Administrator
EvalGoal	Contains records written for every evaluation goal
EvalGoalClassifierItem	Contains records written for every evaluation goal classifier item
EvalGoalClassifierType	Contains records written for every evaluation goal classifier type
EvalGoalEvaluatorSet	Contains records written for every evaluation goal evaluator set
EvalGoalProgress	Contains records written for the progress of evaluation goals
EvalGoalStatus	Contains records written for the status of evaluation goals

Name	Direction
EvalGoalTime	Contains records written for evaluation goal time
EvalGoalType	Contains records written for types of evaluation goals
EvalGoalWhoType	Contains records written for who types of evaluation goals
EvalQuestion	Contains records written for every question scored on an evaluation of a recording
EvalState	Contains records for the possible states of evaluation that a call can have
Evaluator	Contains information about specific evaluators
EvaluatorType	Contains records on the types of administrator roles
EventLogging	Contains information on event logging
EventLoggingParameter	Contains information on the event logging parameter
EventLoggingType	Contains records on the event logging types.
Field	Contains records of valid fields
FieldCategory	Contains records of valid field categories
FileType	Contains records for the possible file types that can be recorded by Quality Management
Filter	Contains records of valid searches
FilterParameter	Contains records of the search filter parameters
FilterProperties	Contains records of the search filter properties.
GamificationConnectionInfo	Contains the valid connection information for gamification
GamificationEvent	Contains the records of events for gamification

Name	Direction
GamificationEventType	Contains the records of event types for gamification
GamificationIcon	Contains the records of icons for gamification
GamificationLevel	Contains the records of levels for gamification
GamificationLevelPointRange	Contains the records of the level point ranges for gamification.
GamificationLevelPointRangePerson	Contains the records of the level point range person for gamification
GamificationLevelScoring	Contains the records of the level scoring for gamification
GamificationPointRange	Contains the records of point ranges for gamification
GamificationPointRangeScoring	Contains the records of point range scoring for gamification
Gateway	Contains configuration information for SPAN recording
GetRandNumber	generates a random number from SQL that is used by Workflow to randomly select recordings
Groups	Contains records written for groups configured in Quality Management Administrator
LicenseBundle	contains records written for the license bundle.
LicenseLibrary	Contains license information. The data in the table is encrypted. Cisco does not publish the encryption key
LoginState	Contains records for the ACD and Quality Management login states
Manager	Contains records written for every person assigned to manage a group configured in Quality Management Administrator

Name	Direction
Media	Contains records of every time an audio or screen recording is uploaded to the Quality Management Record Server.
MediaFile	Contains records of audio and screen media files.
MediaType	Contains information on the types of media
MetaData	Contains records written for every user-defined metadata added to a call in Quality Management in Unified Workforce Optimization® or the Recording API
MetaDataField	Contains records written for every user-defined metadata field configured in Quality Management Administrator
MetaDataMappableColumns	Contains records of every mappable metadata column that is configured in Quality Management Administrator.
MetaDataType	Contains records for the possible types of metadata fields that can be configured in Quality Management Administrator
Path	Contains records written for every path to which a recording is uploaded
Person	Contains records written for every person configured in Quality Management Administrator
PersonField	Contains records with the valid fields for each person
QMAdminAudit	Contains records of the valid QM administrator audit information.
QMAdminAuditAction	Contains records of the valid QM administrator audit action information.
QMAdminAuditArea	Contains records of the valid QM administrator audit area information

Name	Direction
Realm	Contains records for the possible types of realms from which personnel and teams can originate
RealTimeRecordingMonitorState	Contains records of the current recording state
ReconciliationHistory	Contains the records for the reconciliation history
ReconcilationHistoryStatus	Contains the records for the status of the reconciliation history.
RecordingApiCommand	Contains the Recording API command
RecordingApiCommandType	Contains the records that track the Recording API commands sent by users and includes data sent if applicable
RecordingCluster	Contains records written for every recording cluster configured in Quality Management Administrator
RecordingClusterServer	Contains records written for every Record Server configured for a recording cluster in Quality Management Administrator
RecordingEvent	Contains records of recording events
RecordingEventType	Contains records of recording event types
RecordingEventTypeCategory	Contains a record of the recording event type's category
RecordingReason	Contains records for the possible reasons that a call will or will not be recorded for quality or archiving purposes
RecordingStateAudit	Contains the audit information for the recording state
RecordingStateCause	contains records of the failure/success causes for the recording state
RecordingType	Contain records for each recording type
Report	Contains information on each report

Name	Direction
ReportColumn	Identifies the columns in a report
ReportConfiguration	Contains the report configuration information
ReportParameter	Identifies the parameters in a report
ReportRequiredParam	Identifies the required parameters in a report
ReportRole	Identifies the roles associated with a report
ReportRoleScope	Identifies the roles that can view a report
ReportType	Contains records for the possible report types
ReportUserConfig	Identifies who has access to a report
RequiredFilterParam	Identifies the required parameters for an ACD platform
RetentionData	Contains records for each workflow’s retention periods
RetentionType	Contains records for the available retention types
Role	Contains a record for each available role in Quality Management
RtpFilter	Contains the port and IP address information that should be filtered from recording
RtpFilterType	Contains records for the possible RTP filter types
ScreenMonitoring	Contains records written for every time a agent logs into their desktop where Desktop Record service is running
Server	Contains records for each configured server
ServerProperties	Specifies the properties for each configured server
ServerType	Contains records for the possible server types
SignalingGroup	Contains records that specify the type of server for each configured server

Name	Direction
SignalingGroupServer	Contains records that specify the type of server for each configured server
Site	Contains records for each configured site
SiteServer	Contains records that specify which server belongs to a site
Supervisor	Contains records written for every supervisor configured in Quality Management Administrator
Survey	Contains records that specify the type of server for each configured server
SurveyForm	Contains records for each survey form.
SurveyFormStatus	Contains records on the status of each survey form
SurveyQuestion	Contains records on of each question in a survey form.
SurveyResults	Contains records for each survey result.
SyncFilter	Contains records of the survey result filter
SyncFilterACD	Contains records of the sync filter for the ACD
SyncFilterTelephony	Contains the records for the sync filter for telephony
Team	Contains records for every team configured in Quality Management Administrator
TeamGroup	A relationship table between teams and groups
TelephonyGroup	Contains records that provides a unique set of phones and telephony devices. It also includes a number of telephony servers, including at least one Quality Management CTI server
TelephonyGroupServer	A relationship table between servers and telephony groups

Name	Direction
TelephonyGroupType	Contains records for possible telephony group types
Timezone	Contains records for the possible time zones in which the Quality Management is in use
UI_Feature	Contains records for of the valid features
UI_FeatureGroup	Contains a unique set of features groups
UI_FeaturePermissions	Contains the feature permissions for each UI_ Feature based on Role and License
UniqueAdminGroup	Contains records for unique ID numbers used by Quality Management Administrator to keep group number and knowledge worker IDs unique when saving them
UploadState	Contains upload state records for audio and screen recordings
UserReport	Contains information on each user report
UserReportParameters	Identifies the parameters for each user report
VoiPMonitorDevice	Contains records for each configured VoIP device
VoiPMonitorDeviceLine	Contains records that map the devices to extensions and partitions. A device might have multiple extensions and partitions
VoiPMonitorDeviceType	Contains records for each device type associated with a VoiPMonitorDevice
vw_ActiveAgentsInGroups	Creates a new record for active agents in groups
vw_ActiveAgentsInTeams	Creates a new record for active agents in teams
vw_ContactsWithActiveOrg	Contains records used for contact reconciliation
vw_PersonDetails	Creates a new record for person details

Name	Direction
vw_WfmAgentReportCard	Creates a new record for the WFM agent report card
Workflow	Contains records for each configured archive and quality workflow
WorkflowClassifier	Contains records for each workflow classifier
WorkflowClassifierNumber	Contains records for each workflow classifier number
WorkflowClassifierNumberType	Contains records for each workflow classifier number
WorkflowClassifierType	Contains records for each workflow classifier type
WorkflowRule	Contains records for each workflow classifier rule
WorkflowRuleWhat	Contains records for each WHAT rule in a workflow
WorkflowRuleWhatPeriod	Contains records for each period in a WHAT rule
WorkflowRuleWhenRange	Contains records for each range in a WHEN rule
WorkflowRuleWhenType	Contains records for each type in a WHEN rule
WorkflowRuleWhenWeekly	Contains records for each WHEN rule with a Select When of Weekly
WorkflowRuleWhenWeeklyType	Contains records for the day of the week
WorkflowRuleWhoPerson	Contains records for each a person associated with a WHO rule
WorkflowRuleWhoTeam	Contains records for each team associated with a WHO rule
WorkflowRuleWhoType	Contains records for each type for a WHO rule in a workflow
WorkflowType	Contains the available types for a workflow (that is, quality and archive)

ACD

The ACD table contains information about the Automatic Call Distributor (ACD).

Column Name	Description	Storage
id	Numeric identifier for the ACD, auto-generated by the database.	int identity NOT NULL Primary Key
name	Identifies the name of the ACD.	varchar[256] NOT NULL
acdTypeFK	A foreign key to the ID in the acdType table.	int NOT NULL
doReconciliation	0–the ACD is marked for reconciliation. 1–the ACD is not marked for reconciliation.	bit NOT NULL
doDataSync	0–the ACD is marked for data sync. 1–the ACD is not marked for data sync.	bit NOT NULL
activated	The GMT date and time when the ACD was configured.	datetime (16,3) NOT NULL
deactivated	The GMT date and time when the ACD was removed.	datetime (16,3) NOT NULL
lastModified	The last time the ACD was modified.	datetime (16,3) NOT NULL

Every ACD record is associated with the following records:

- ACDQueue
- ACDServer
- ACDType
- ReconciliationHistory
- SyncFilterACD
- Team
- Person

ACDQueue

The ACDQueue table contains information about the ACD Queue.

Column Name	Description	Storage
id	Numeric identifier for this alert.	bigint identity NOT NULL
keyParam	The key parameter for this ACD.	int NOT NULL
name	Identifies the name of the ACD.	varchar[64] NOT NULL
extension	The extension for this ACD.	varchar(32) NOT NULL
type	Identifies the type of ACD.	varchar[32] NOT NULL
acdFK	A foreign key to the ID in the ACD table.	int NOT NULL

Every ACDQueue record is associated with the following record:

- ACD

ACDServer

The ACDServer table contains information about the ACD server.

Column Name	Description	Storage
acdFK	A foreign key to the ID in the ACD table.	int NOT NULL
serverFK	A foreign key to the ID in the Server table.	bigint NOT NULL
lastModified	The last time the ACD server was modified.	datetime (16,3) NOT NULL

Every ACDServer record is associated with the following records:

- ACD
- Server

ACDType

The ACDType table contains information about the ACD server.

Column Name	Description	Storage
id	Numeric identifier for the ACD type, auto-generated by the database.	int identity NOT NULL Primary Key
name	Identifies the name of the ACD type.	varchar[32] NOT NULL
lastModified	The last time the ACD type was modified.	datetime (16,3) NOT NULL

Every ACDType record is associated with the following record:

- ACD

Agent

The Agent table contains information about specific agents/knowledge workers. A record is created when a knowledge worker role is assigned in Quality Management Administrator or when an agent is synced from the ACD. After being synced or saved in Quality Management Administrator, it will be at most ten minutes until the Monitoring and Recording Sync service adds this record to the database. Activation and deactivation dates reflect the time of the database sync that created or deactivated the record.

Column Name	Description	Storage
id	Numeric identifier for the agent, auto-generated by the database.	int identity NOT NULL Primary Key
personFK	A foreign key to the ID in the Person table.	int NOT NULL
activated	The GMT date and time when the agent or knowledge worker role was assigned to the person.	datetime (16,3) NOT NULL

Column Name	Description	Storage
deactivated	The GMT date and time when the agent or knowledge worker role was removed from the person.	datetime (16,3) NOT NULL
lastModified	The last time the agent was modified.	datetime (16,3) NOT NULL

Every Agent record is associated with the following records:

- Person
- AgentTeam
- Supervisor

AgentTeam

Quality Management creates an AgentTeam record every time an agent is assigned to a team in Quality Management Administrator or when an agent on a team is synced from the ACD.

After being synced or saved in Quality Management Administrator, it will be at most ten minutes until the Monitoring and Recording Sync service adds this record to the database. Activation and deactivation dates reflect the time of the database sync that created or deactivated the record.

Column Name	Description	Storage
agentFK	A foreign key to the ID in the Agent table.	int NOT NULL
teamFK	A foreign key to the ID in the Team table.	int NOT NULL
activated	The GMT date and time when the agent or knowledge worker role was assigned to the team.	datetime (16,3) NOT NULL
deactivated	The GMT date and time when the agent or knowledge worker role was removed from the team.	datetime (16,3) NOT NULL
lastModified	The last time the agent team was modified.	datetime (16,3) NOT NULL

Every AgentTeam record is associated with the following records:

- Team
- Agent

Alert

The Alert table contains a record for each user alert. Quality Management generates alerts when a significant event happens, and sends them to agents, evaluators, supervisors, and managers.

Column Name	Description	Storage
id	Numeric identifier for this alert.	bigint identity NOT NULL Primary Key
personFK	A foreign key to the ID in the Person table.	int NOT NULL
date	The date when the message was generated	datetime (16,3) NOT NULL
message	A description of the alert.	nvarchar[1000] NOT NULL
summary	A brief description of the alert.	nvarchar[512]
state	Identifies any error condition associated with the alert. The possible states are: 0—Idle 1—Info Ready 2—Info 3—Warning 4—Error 5—Fatal Error	nvarchar[64]
type	Identifies the name of the alert type. The type is related to name in the AlertType table. However, since Alerts can be sent by any application (even third parties), this is not a foreign key because the alert system might not use all alert types that are listed under name in the AlertType table. The value in this column might also include alert types that are not listed in the AlertType table.	nvarchar[64]

Every Alert record is associated with the following record:

- Person

AlertType

The AlertType table contains a record for each alert type.

Column Name	Description	Storage
id	Numeric identifier for this alert type.	int identity NOT NULL Primary Key
name	Identifies the name of the alert type.	varchar[64] NOT NULL

Every AlertType record is associated with the following record:

- EvalAlerts

ApprovalType

The ApprovalType table contains information about the approval type.

Column Name	Description	Storage
id	Numeric identifier for the approval type, auto-generated by the database.	int identity NOT NULL Primary Key
name	Identifies the name of the approval type.	varchar[32] NOT NULL

Every ApprovalType record is associated with the following record:

- EvalForm

ArchiveAudit

Quality Management creates a new record in the ArchiveAudit table every time an audio recording is played in the Search and Play widget in Unified Workforce Optimization.

Column Name	Description	Storage
ccrFK	A foreign key to the ID in the Ccr table.	bigint NOT NULL
personFK	A foreign key to the ID in the Person table. This is the person who played the recording.	int NOT NULL

Column Name	Description	Storage
accessed	The date and time playback of the contact started in the Evaluate and Review widget.	datetime (16,3) NOT NULL
accessedTzFK	A foreign key into the Timezone table. A value of zero indicates that the time zone is currently unknown. An unknown time zone might be updated at some point. For more information, see Dates in the Database.	tinyint NOT NULL
accessedTzOffset	Time zone Offset from GMT; a value representing a positive or negative offset in quarter-hourly intervals from GMT (UTC). Adding (TzOffset - 128) x 15 minutes to the corresponding (GMT) timestamp will always provide the local time. For more information, see Dates in the Database.	tinyint NOT NULL

Every ArchiveAudit record is associated with the following records:

- Ccr
- Person
- Timezone

AxlUCCMSchema

The AxlUCCMSchema associates a version of Cisco Unified CM with an Administration XML (AXL) schema version in a telephony group.

Column Name	Description	Storage
id	Numeric identifier for the Ax1UCCMSchema, auto-generated by the database.	bigint identity NOT NULL Primary Key
uccmVersion	The Unified CM version.	varchar[32] NOT NULL
schemaVersion	The AXL schema version.	varchar[32] NOT NULL

Ccr

Quality Management creates a new record in the Ccr table for each call or call leg, known as a Customer Contact Record (Ccr) processed by the system. A new call leg starts each time that a call is transferred or redirected.

A Ccr record contains detailed information about the call or leg. At least one such record exists for each call.

Column Name	Description	Storage
id	Numeric identifier for this customer contact record.	bigint identity NOT NULL Primary Key
ccrTypeFK	A foreign key to the ID in the CcrType table.	int NOT NULL
startTime	The date and time this contact began, in GMT.	datetime (16,3) NOT NULL
startTimeTzFK	A foreign key into the Timezone table. A value of zero indicates that the time zone is currently unknown. An unknown time zone might be updated at some point. For more information, see Dates in the Database.	tinyint NOT NULL
startTimeTzOffset	Time zone Offset from GMT; a value representing a positive or negative offset in quarter-hourly intervals from GMT (UTC). Adding (TzOffset - 128) x 15 minutes to the corresponding (GMT) timestamp will always provide the local time. For more information, see Dates in the Database.	tinyint NOT NULL
localStartTime	The local start time of the call in the timezone where the call occurred. The localStartTime is used by Quality Management Reporting to compare the start time with the local date range selections when viewing reports.	datetime (16,3) NOT NULL
offset	The time the recording starts relative to the startTime timestamp of the Ccr, in milliseconds. A positive value indicates the recording starts after the startTime and negative value indicates that it starts before.	int NOT NULL
duration	The duration of the call from time answered to time dropped, in milliseconds.	int NOT NULL
recordingTypeFK	A foreign key to the RecordingType table.	int NOT NULL

Column Name	Description	Storage
siteFK	A foreign key to the Site table.	int NOT NULL
associatedCallId	The ID to associate related contacts. This is used to display associated calls in the Quality Management in Unified Workforce Optimization, so various legs of the same customer contact can be viewed together.	varchar[52] NOT NULL
isReconciled	0—the Gateway Recording is not reconciled. 1—the Gateway Recording is reconciled.	bit NOT NULL
icmCallId	The Peripheral Call Key (or Call ID) for the call from the CCM system.	varchar(128) NOT NULL
ani	The originating phone number for this contact.	nvarchar(128) NOT NULL
dnis	The called phone number for this contact.	nvarchar(128) NOT NULL
line	Agent's extension.	nvarchar(128) NOT NULL
wasAnswered	1—the Ccr was answered. 0—the Ccr was not answered.	bit NOT NULL
timeToAnswer	Milliseconds from the startTime until the agent answered the incoming call.	int NOT NULL
personFK	A foreign key to the ID in the Person table.	int NOT NULL
evalFormFK	A foreign key reference to ID in the EvalForm table.	int NOT NULL
evaluatorFK	A foreign key to the ID in the Person table. This is the person who evaluated the Ccr or claimed it for evaluation, or null if it has not yet been claimed or evaluated.	int NOT NULL
evalStateFK	A foreign key to the ID in the EvalState table.	tinyint NOT NULL

Column Name	Description	Storage
isCalibration	1–the contact recording was marked for calibration. 0–the contact recording was not marked for calibration.	bit NOT NULL
calibrationFormFK	A foreign key to the ID in the calibrationForm table.	int NOT NULL
isTraining	1–the Ccr was marked for training during the evaluation process. 0–the Ccr was not marked for training during the evaluation process.	bit NOT NULL
isPip	1–the Ccr was marked for Human Resources during the evaluation process. 0–the Ccr was not marked for Human Resources during the evaluation process.	bit NOT NULL
qualityWorkflowFK	A foreign key into the Workflow table.	int
archiveWorkflowFK	A foreign key into the Workflow table.	int
archiveReasonFK	A foreign key to the ID in the RecordingReason table. The reason why the call was or was not marked for archiving.	smallint NOT NULL
qualityReasonFK	A foreign key to the ID in the RecordingReason table. The reason why the call was or was not marked for quality management purpose.	smallint NOT NULL
audioUploadState	A foreign key to the ID in the UploadState table.	tinyint NOT NULL
screenUploadState	A foreign key to the ID in the UploadState table.	tinyint NOT NULL
isInbound	1–the call is inbound.� 0–the call is not inbound.	bit NOT NULL
teamFK	A foreign key to the ID in the Team table.	int NOT NULL
groupFK	A foreign key to the ID in the Groups table.	int NOT NULL

Column Name	Description	Storage
fromAddress	An address in the From field of a non-call contact.	varchar[512] NOT NULL
toAddress	An address in the To field of a non-call contact.	varchar[512] NOT NULL
subject	The text in the Subject field of a non-call contact.	varchar[1996] NOT NULL
telephonyGroupFK	A foreign key to the ID in the telephonyGroup table.	int NOT NULL
audioFK	A foreign key to the ID in the Audio table.	bigint NOT NULL
screenFK	A foreign key to the ID in the Screen table.	bigint NOT NULL
licenseBundleFK	A foreign key to the licenseBundle table.	int NOT NULL
lastModified	The last time the Ccr was modified.	datetime (16,3) NOT NULL

Every Ccr record is associated with the following records:

- ArchiveAudit
- Cdr
- CdrCcr
- CcrType
- Eval
- EvalForm
- EvalGoalProgress
- EvalState
- Groups
- LicenseBundle

- Media
- MetaData
- Person
- ReconciliationHistory
- RecordingEvent
- RecordingReason
- RecordingType
- Site
- Team
- TelephonyGroup
- Timezone
- UploadState
- Workflow

CcrType

The CcrType specifies the type of contact as either call or non-call.

Column Name	Description	Storage
id	Numeric identifier for this customer contact record.	int NOT NULL Primary Key
name	The name of the CcrType.	varchar [2147483647]

Every CcrType record is associated with the following records:

- Ccr—Every CcrType record is associated with a Ccr record.

Cdr

Quality Management creates a new record in the call detail record (Cdr) table for each associated contact. The information is pulled from the Cisco call detail records. See *Cisco Unified Communications Manager Call Detail Records Administration Guide* for additional information.

Column Name	Description	Storage
pkid	The partition name that is associated with the OriginalCalled PartyNumber.	varchar[100] NOT NULL
globalCallId_callManagerId	A unique Cisco Unified CM identity.	int NOT NULL
globalCallID_callId	A unique call identity value that is assigned to each call.	int NOT NULL
origLegCallIdentifier	The originating leg of a call.	int NOT NULL
destLegIdentifier	The terminating leg of a call.	int NOT NULL
dataTimeOrigination	The date and time when the user goes off hook or the date and time that the H.323 SETUP message is received for an incoming call. The time is stored as UTC.	int NOT NULL
dateTimeConnect	The date and time that the call connects. If the call is never answered, this value shows zero. The time is stores as UTC. The default value is 0.	int NOT NULL
dateTimeDisconnect	The date and time when the call is cleared. This field is set even if the call never connects. The time is stores as UTC. The default value is 0.	int NOT NULL
duration	The duration of the recording event in milliseconds. The value is derived as the difference between the dateTimeConnect and dateTimeDisconnect. The field remains zero if the call never connects of if it connects for less than 1 second. The default value is 0.	int NOT NULL

Column Name	Description	Storage
callingPartyNumber	The phone number that made the call.	varchar[50] NOT NULL
originalCalledPartyNumber	The phone number that first received the call. If a blended address is used, this field contains the directory number portion of the blended address. This field represents an alphanumeric string that can be either digits or a SIP URL. If a destination cannot be reached, this field is empty.	varchar[50] NOT NULL
finalCalledPartyNumber	The phone number that last received the call. If no forwarding occurs, this number shows the same number as the originalCalledPartyNumber. If the call is presented to a directory URI, the field remains empty. If a blended address is used, this field contains the directory number portion of the blended address. For calls to a conference bridge, this field contains the actual identifier of the conference bridge, which is an alphanumeric string. This field represents an alphanumeric string that can be either digits or a SIP URL. If a destination cannot be reached, this field is empty.	varchar[100] NOT NULL

Column Name	Description	Storage
lastRedirectDN	For forwarded calls, this field specifies the phone number of the next to last hop before the call reaches its final destination. If only one hop occurs, this number matches the OriginalCalledPartyNumber. If a blended address is used, this field contains the directory number portion of the blended address. For calls that are not forwarded, this field displays the OriginalCalledPartyNumber and the FinalCalledPartyNumber. For calls to a conference bridge, this field contains the actual identifier of the conference bridge, which is an alphanumeric string.	varchar[50] NOT NULL
huntPilotDN	The hunt pilot domain name (DN) through which the call is routed.	varchar[50] NOT NULL
origDeviceName	The name of the originating device.	varchar[129] NOT NULL
destDeviceName	The name of the destination device. If the device does not have a name the field remains empty.	varchar[129] NOT NULL
IncomingProtocolCallRef	A globally unique call reference identification for the protocol.	varchar[45] NOT NULL
OutgoingProtocolCallRef	The protocol used between Cisco Unified CM and the downstream voice product in the call path.	varchar[45] NOT NULL
finalCalledPartyUnicodeLoginUserID	The unicode login user ID for the final called party in a conference call.	varchar[50] NOT NULL
callingPartyNumberPartition	The number partition for the calling party in a conference call.	varchar[50] NOT NULL

Column Name	Description	Storage
finalCalledPartyNumberPartition	The number partition for the final called party number in a conference call.	varchar[50] NOT NULL
origConversationId	The originator conversation ID in a conference call.	int NOT NULL
destConversationId	The destination conversation ID in a conference call.	int NOT NULL

CdrCcr

The CdrCcr associates Cisco CDR records with QM CCR records.

Column Name	Description	Storage
pkidFK	A foreign key to the pkid in the Cdr table.	nvarchar[100] NOT NULL
ccrFK	A foreign key to the ID in the Ccr table.	bigint NOT NULL

Every CdrCcr record is associated with the following record:

- Ccr

DashRoleView

The DashRoleView table associates a dashboard view with a role.

Column Name	Description	Storage
dashViewFK	A foreign key to the ID in the DashView table.	int NOT NULL
roleFK	A foreign key to the ID in the Role table.	smallint NOT NULL

Every DashRoleView record is associated with the following records:

- DashView
- Role

DashUserView

Quality Management loads the DashUserView table with the valid Dashboard views for each user.

Column Name	Description	Storage
personFK	A foreign key to the ID in the Person table.	int NOT NULL
dashViewFK	A foreign key to the DashView table.	int NOT NULL
userViewId	The view ID for the user.	int NOT NULL
isCurrent	1–if this user is the current user. 0–if this user is not the current user.	bit NOT NULL
id	Numeric identifier for this Dashboard user view. This identifier is auto-generated by the database.	int identity NOT NULL Primary Key

Every DashUserView record is associated with the following records:

- Person
- DashView

DashView

Quality Management loads the DashView table with the valid Dashboard views.

Column Name	Description	Storage
id	Numeric identifier for this Dashboard view. This identifier is auto-generated by the database.	int identity NOT NULL Primary Key
name	Name of this Dashboard view. Names must be unique.	nvarchar(100) NOT NULL
isReadOnly	1–if this Dashboard view is read only. 0–if this Dashboard view is not read only.	bit NOT NULL

Column Name	Description	Storage
isAdminCreated	1–if this Dashboard view is created by the administrator. 0–if this Dashboard view is not created by the administrator and is created by a user.	bit NOT NULL

Every DashView record is associated with the following records:

- DashRoleView
- DashUserView
- DashWidgetView

DashWidget

Quality Management loads the DashWidget table with the valid Dashboard widgets.

Column Name	Description	Storage
id	Numeric identifier for this Dashboard gadget. This identifier is auto-generated by the database.	int identity NOT NULL Primary key
name	Name of this Dashboard gadget.	varchar(50) NOT NULL
contentPath	The URL location where the widget resides.	varchar(1000) NOT NULL

Every DashWidget record is associated with the following records:

- DashWidgetRole
- DashWidgetSetting
- DashWidgetView

DashWidgetRole

Quality Management loads the DashWidgetRole table with the valid roles for each widget.

Column Name	Description	Storage
roleFK	A foreign key to the ID in the Role table.	smallint NOT NULL
dashWidgetFK	A foreign key to the ID in the DashWidget table.	int NOT NULL

Every DashWidgetRole record is associated with the following records:

- Role
- DashWidget

DashWidgetSetting

Quality Management loads the DashWidgetSetting table with the valid settings for each widget.

Column Name	Description	Storage
id	Numeric identifier for this Dashboard gadget setting. This identifier is auto-generated by the database.	int identity NOT NULL Primary key
dashWidgetFK	A foreign key to the ID in the DashWidget table.	int NOT NULL
name	Name of this Dashboard gadget setting.	varchar(50) NOT NULL
defaultValue	The default value for this Dashboard gadget setting.	nvarchar (1073741823) NOT NULL
isLockable	1—if this Dashboard view is lockable by the administrator. 0—if this Dashboard view is not lockable by the administrator.	bit NOT NULL

Every DashWidgetSetting record is associated with the following records:

- DashWidget
- DashWidgetViewSetting

DashWidgetView

Quality Management loads the DashWidgetView table with the valid views for each widget.

Column Name	Description	Storage
id	Numeric identifier for this Dashboard gadget view. This identifier is auto-generated by the database.	int identity NOT NULL Primary Key
dashWidgetFK	A foreign key to the ID in the DashWidget table.	int NOT NULL
dashViewFK	A foreign key to the ID in the DashView table.	int NOT NULL
name	Name of this Dashboard gadget view.	nvarchar(100) NOT NULL
position	Location of the Dashboard gadget view in the Dashboard grid.	int NOT NULL

Every DashWidgetView record is associated with the following records:

- DashWidget
- DashView
- DashWidgetViewSetting

DashWidgetViewSetting

Quality Management loads the DashWidgetViewSetting table with the valid settings for each Dashboard view.

Column Name	Description	Storage
id	Numeric identifier for this Dashboard gadget (widget) view setting. This identifier is auto-generated by the database.	int identity NOT NULL Primary Key
dashWidgetViewFK	A foreign key to the ID in the DashWidgetView table.	int NOT NULL

Column Name	Description	Storage
dashViewSettingFK	A foreign key to the ID in the DashViewSetting table.	int NOT NULL
value	The Dashboard gadget view setting.	nvarchar (1073741823) NOT NULL
personFK	A foreign key to the ID in the Person table.	int NOT NULL

Every DashWidgetViewSetting record is associated with the following records:

- DashWidgetView
- DashWidgetSetting
- Person

DbProperties

Quality Management adds a record for various general quality management settings after the schema is created. This is a generic table where the data is somewhat dynamic depending on the version of the Quality Management. The database properties in this version are:

- ArchiveFileType: 3
- MetadataKey: customer specific metadata encryption key to use for decrypting metadata
- SchemaMajorVersion: current major version of the database schema
- SchemaMinorVersion: current minor version of the database schema

Note: MetadataKey will only be added to this table if and when encrypted Metadata Fields are being used in Quality Management.

Column Name	Description	Storage
id	Numeric identifier for this database property.	varchar(32) NOT NULL Primary Key

Column Name	Description	Storage
setting	The database property setting.	nvarchar(2048) NOT NULL

Eval

Quality Management adds a record to the Eval table each time a Ccr is claimed for evaluation. This record is added the first time the evaluation is saved, whether it is just claimed or scored partially or fully.

A Ccr record contains detailed information about the call or leg. At least one such record exists for each call.

Column Name	Description	Storage
id	Numeric identifier for this evaluation. This number is auto-generated by the database.	int identity NOT NULL Primary Key
ccrFK	A foreign key to the ID in the Ccr table.	bigint NOT NULL
evalFormFK	A foreign key to the ID in the EvalForm table.	int NOT NULL
totalScore	The score of this evaluation. Null if the evaluation has been claimed but no questions have yet been scored.	float NULL
evaluated	The date and time this evaluation was last saved, in GMT.	datetime (16,3) NOT NULL
evaluatedTzFK	A foreign key into the Timezone table. A value of zero indicates that the time zone is currently unknown. An unknown time zone might be updated at some point. For more information, see Dates in the Database.	tinyint NOT NULL
evaluatedTzOffset	Time zone Offset from GMT; a value representing a positive or negative offset quarter-hourly intervals from GMT (UTC). Adding (TzOffset - 128) x 15 minutes to the corresponding (GMT) timestamp will always provide the local time. For more information, see Dates in the Database.	tinyint NOT NULL

Column Name	Description	Storage
approverFK	A foreign key to the ID of the person in the Person table. The person who approved the evaluation if the evaluation form used required approval. Column Name is null if the evaluation form used for this evaluation does not require approval or if the required approval has not yet been done.	int NULL
evalStateFK	A foreign key to the ID in the EvalState table.	tinyint NOT NULL
isCountedScore	1–if the score is counted. 0–if the score is not counted.	bit NOT NULL
evaluatorFK	A foreign key to the ID in the Person table. This is the person who evaluated the Ccr or claimed it for evaluation, or null if it has not yet been claimed or evaluated.	int NOT NULL
additiveScore	The points for this option are added together to get the final score, assuming no Key Performance Indicator (KPI) options are selected.	int NOT NULL
lastModified	The last time the Eval was modified.	datetime (16,3) NOT NULL

Every Eval record is associated with the following records:

- Ccr
- EvalForm
- EvalQuestion
- EvalComment
- EvalState
- Person
- Timezone
- EvalComment
- EvalState
- Ccr

- Timezone
- Person

EvalAlerts

The EvalAlerts table contains a record for each evaluation alert. Quality Management generates evaluation alerts when changes are made to an evaluation form.

Column Name	Description	Storage
evalFormFK	A foreign key to the ID in the EvalForm table.	int NOT NULL Primary key
roleFK	A foreign key to the ID in the Role table.	smallint NOT NULL Primary key
alertFK	A foreign key to the ID in the AlertType table.	int NOT NULL Primary key
enabled	1–if users with a specified role can receive the specified alert for the evaluation form. 0–if users with a specified role cannot receive the specified alert for the evaluation form.	bit NOT NULL

Every EvalAlerts record is associated with the following records:

- EvalForm
- Role
- AlertType

EvalComment

Quality Management adds a record to the EvalComment table each time a comment is added to a particular evaluation of a recording.

Column Name	Description	Storage
id	Numeric identifier for the evaluation comment, auto-generated by the database.	int identity NOT NULL Primary Key
evalFK	A foreign key to the ID in the Eval table.	int NOT NULL
personFK	A foreign key to the ID of the person who made the comment, in the Person table.	int NOT NULL
text	The text of the comment.	ntext NOT NULL
created	The date and time the comment was created and saved, in GMT.	datetime (16,3) NOT NULL
createdTzFK	A foreign key into the Timezone table. A value of zero indicates that the time zone is currently unknown. An unknown time zone might be updated at some point. For more information, see Dates in the Database.	tinyint NOT NULL
createdTzOffset	Time zone Offset from GMT; a value representing a positive or negative offset in quarter-hourly intervals from GMT (UTC). Adding (TzOffset - 128) x 15 minutes to the corresponding (GMT) timestamp will always provide the local time. For more information, see Dates in the Database.	tinyint NOT NULL
audioRecordingOffset	The start of an event as an offset from the beginning of the audio **Example:** An offset appears where a comment occurred.	int NOT NULL
parentCommentFK	The location where the comment originated.	int NOT NULL
sectionFK	A foreign key to the ID in the EvalFormSection table.	int NOT NULL

Column Name	Description	Storage
questionFK	A foreign key to the ID in the EvalFormQuestion table.	int NOT NULL

Every EvalComment record is associated with the following records:

- Eval
- EvalFormQuestion
- EvalFormSection
- Person
- Timezone

EvalForm

Quality Management adds a record to the EvalForm table each time a new evaluation form is created. The system adds defaults to this table for the four evaluation form templates that are installed with Quality Management (one blank template and two templates with sections and questions).

Column Name	Description	Storage
id	Numeric identifier for this evaluation form. This identifier is auto-generated by the database.	int identity NOT NULL Primary Key
name	Name of this evaluation form. Names must be unique.	nvarchar (128) NOT NULL
description	Description of this evaluation form.	nvarchar (256) NOT NULL
creator	Creator of the evaluation form as entered in the Original Author field in Quality Management Administrator. The default value in this field is Administrator. This is the value in the evaluation form templates. It can be changed when you create an evaluation form.	nvarchar (128) NOT NULL

Column Name	Description	Storage
created	Date and time this evaluation form was created, in GMT.	datetime (16,3) NOT NULL
updater	Updater of this evaluation form as entered in the Last Author field of Quality Management Administrator. For the default templates this is set to Administrator.	nvarchar (128) NOT NULL
updated	Date and time this evaluation form was updated, in GMT.	datetime (16,3) NOT NULL
isDefaultForm	1–if this evaluation form is the default. 0–if this evaluation form is not the default.	bit NOT NULL
status	The status of the evaluation form. The possible values are: 0–Editable; It is created and edited by an administrator. 1–Active; Editing is complete; can be used for evaluation; can no longer be deleted without affecting scored evaluations. 2–Inactive; Cannot be used for evaluating and will be deleted when all its evaluations are deleted and 13 months have passed. 3–Template; Used to create new evaluation forms but not used for evaluating.	tinyint NOT NULL
bandMax1	The maximum percentage value for the “Needs Improvement” scoring band.	int NOT NULL
bandMax2	The maximum percentage value for the “Meets Expectations” scoring band.	int NOT NULL
createdTzFK	A foreign key into the Timezone table. A value of zero indicates that the time zone is currently unknown. An unknown time zone might be updated at some point. For more information, see Dates in the Database.	tinyint NOT NULL

Column Name	Description	Storage
createdTzOffset	Time zone Offset from GMT; a value representing a positive or negative offset in quarter-hourly intervals from GMT (UTC). Adding (TzOffset - 128) x 15 minutes to the corresponding (GMT) timestamp will always provide the local time. For more information, see Dates in the Database.	tinyint NOT NULL
updatedTzFK	A foreign key into the Timezone table. A value of zero indicates that the time zone is currently unknown. An unknown time zone might be updated at some point. For more information see Dates in the Database.	tinyint NOT NULL
updatedTzOffset	Time zone offset from GMT; a value representing a positive or negative offset in quarter-hourly intervals from GMT (UTC). Adding (TzOffset - 128) x 15 minutes to the corresponding (GMT) timestamp will always provide the local time. For more information, see Dates in the Database.	tinyint NOT NULL
evaluatorsCanApprove	1–if evaluators can approve evaluations scored with this evaluation form. 0–if evaluators cannot approve evaluations scored with this evaluation form.	bit NOT NULL
managersCanApprove	1–if managers can approve evaluations scored with this evaluation form. 0–if managers cannot approve evaluations scored with this evaluation form.	bit NOT NULL
supervisorsCanApprove	1–if supervisors can approve evaluations scored with this evaluation form. 0–if supervisors cannot approve evaluations scored with this evaluation form.	bit NOT NULL
canEditComments	1–if users can edit comments. 0–if users cannot edit comments.	bit NOT NULL

Column Name	Description	Storage
scoringTypeFK	A foreign key into the EvalFormScoringType table.	tinyint NOT NULL
approvalTypeFK	A foreign key into the approvalType table.	int NOT NULL
selfEvaluationNeedsCalibration	1–if self evaluation needs calibration. 0–if self evaluation does not need calibration.	bit NOT NULL
lastModified	The last time the EvalForm was modified.	datetime (16,3) NOT NULL

Every EvalForm record is associated with the following records:

- Ccr
- Eval
- EvalAlerts
- ApprovalType
- EvalFormScoringType
- EvalFormSection
- Timezone
- WorkflowClassifier

EvalFormQuestion

Quality Management adds a record to the EvalFormQuestion table each time a new question in an evaluation form section is created. The system adds defaults to this table for the evaluation form templates that are installed with Quality Management.

Column Name	Description	Storage
id	Numeric identifier for this evaluation form question. This number is auto-generated by the database.	int identity NOT NULL Primary Key

Column Name	Description	Storage
sectionFK	A foreign key to the ID in the EvalFormSection table.	int NOT NULL
text	The text for this question.	nvarchar (1073741823) NOT NULL
ordinal	The order number for this question. 1—if the question is a key performance indicator (KPI) question. 0—if the question is not a KPI question.	tinyint NOT NULL
weight	The weight of this question. Max = 1.0.	float NOT NULL
description	An explanation of the question. The description can contain a maximum of 500 characters. The description should explain the scoring criteria used to keep evaluations in calibration.	nvarchar (1072741823) NOT NULL
lastModified	The last time the EvalFormQuestion was modified.	datetime (16,3) NOT NULL

Every EvalFormQuestion record is associated with the following records:

- EvalComment
- EvalFormSection
- EvalQuestion
- EvalQuestionOption

EvalFormQuestionOption

Quality Management adds a record to the EvalFormQuestionOption table each time an option is configured for a question.

Column Name	Description	Storage
id	Numeric identifier for this evaluation form question option. This number is auto-generated by the database.	int identity NOT NULL Primary Key

Column Name	Description	Storage
questionFK	A foreign key to the ID in the EvalFormQuestion table.	int NOT NULL
typeFK	A foreign key into the EvalFormQuestionOptionType table.	tinyint NOT NULL
label	The text for the question option.	nvarchar(128) NOT NULL
ordinal	The order number for this question option.	int NOT NULL
isDefault	1–if this question option is the default. 0–if this question option is not the default.	bit NOT NULL
kpiPriority	Indicates whether this question template option is a KPI answer.	int NOT NULL
points	The point value assigned to the question option.	int NOT NULL
lastModified	The last time the EvalFormQuestionOption was modified.	datetime (16,3) NOT NULL

Every EvalFormQuestionOption record is associated with the following records:

- EvalFormQuestion
- EvalFormQuestionOptionType
- EvalQuestion

EvalFormQuestionOptionType

Quality Management adds a record to the EvalFormQuestionOptionType table each time a new option type for a question is created.

Column Name	Description	Storage
id	Numeric identifier for this question option type.	int identity NOT NULL Primary Key
name	The name of the question option type.	varchar[32]

Every EvalFormQuestionOptionType record is associated with the following records:

- EvalFormQuestionOption
- EvalFormQuestionTemplateOption

EvalFormQuestionTemplate

Quality Management adds a record to the EvalFormQuestionTemplate table each time a template is created.

Column Name	Description	Storage
id	Numeric identifier for this evaluation form question template.	int identity NOT NULL Primary Key
name	The name of the evaluation form question template.	nvarchar[128]

Every EvalFormQuestionTemplate record is associated with the following records:

- EvalFormQuestionTemplateOption

EvalFormQuestionTemplateOption

Quality Management adds a record to the EvalFormQuestionTemplateOption table each time a template option is selected.

Column Name	Description	Storage
id	Numeric identifier for this evaluation form question template option.	int identity NOT NULL Primary Key
templateFK	The name of the evaluation form question template.	int NOT NULL
typeFK	A foreign key into the EvalFormQuestionOptionType table.	tinyint NOT NULL
label	The text for this question template option.	nvarchar(128) NOT NULL

Column Name	Description	Storage
ordinal	The order number for this question template option. 1—if the question is a key performance indicator (KPI) question. 0—if the question is not a KPI question.	int NOT NULL
isDefault	The default answer for this question template option. 1—if this question template option is the default. 0—if this question template option is not the default.	bit NOT NULL
kpiPriority	Indicates whether this question template option is a KPI answer.	int NOT NULL
points	The point value assigned to the question template option.	int NOT NULL

Every EvalFormQuestionTemplateOption record is associated with the following records:

- EvalFormQuestionTemplate
- EvalFormQuestionOptionType

EvalFormScoringType

Quality Management adds a record to the EvalFormScoringType table each time a new scoring type is created.

Column Name	Description	Storage
id	Numeric identifier for this scoring type.	int identity NOT NULL Primary Key
name	The type of scoring. The scoring type can be points-based or percentage-based.	varchar[32]

Every EvalFormScoringType record is associated with the following records:

- EvalForm

EvalFormSection

Quality Management adds a record to the EvalFormSection table, each time a new section of an evaluation form is created. The system adds defaults to this table for the evaluation form templates that are installed with Quality Management.

Column Name	Description	Storage
id	Numeric identifier for this evaluation form section, auto-generated by the database.	int identity NOT NULL Primary Key
evalFormFK	A foreign key to the ID in the EvalForm table.	int NOT NULL
name	Name of this evaluation form section. This must be unique for the same evalFormFK.	nvarchar(128) NOT NULL
ordinal	The order number for this section.	tinyint NOT NULL
weight	The weight of this section. Max = 1.0.	float NOT NULL
startColor	The starting gradient color for the chart results in a points-based evaluation form. If None is chosen, the section will not appear in the chart.	varchar[8] NOT NULL
endColor	The ending gradient color for the chart results in a points-based evaluation form. The ending gradient color is predefined based on the starting gradient color.	varchar[8] NOT NULL
lastModified	The last time the EvalFormSection was modified.	datetime (16,3) NOT NULL

Every EvalFormSection record is associated with the following records:

- EvalForm
- EvalFormQuestion
- EvalComment

EvalGoal

The EvalGoal table contains information about the evaluation goal.

Column Name	Description	Storage
id	Numeric identifier for the evaluation goal, auto-generated by the database.	int identity NOT NULL Primary Key
name	Identifies the name of the evaluation goal.	varchar[510] NOT NULL
goalQuantity	The number of goals.	int NOT NULL
evalTypeFK	A foreign key to the ID in the evalType table.	tinyint NOT NULL
whoTypeFK	A foreign key to the ID in the EvalGoalWhoType table.	tinyint NOT NULL
timeFK	A foreign key to the ID in the time table.	tinyint NOT NULL
statusFK	A foreign key into the status table.	tinyint NOT NULL
doDataSync	0–the ACD is marked for data sync. 1–the ACD is not marked for data sync.	bit NOT NULL
deactivated	The GMT date and time when the evaluation goal was removed.	datetime (16,3) NOT NULL
activated	The GMT date and time when the evaluation goal was configured.	datetime (16,3) NOT NULL
lastModified	The last time the evaluation goal was modified.	datetime (16,3) NOT NULL

Every EvalGoal record is associated with the following records:

- EvalGoalClassifierItem
- EvalGoalEvaluatorSet
- EvalGoalProgress
- EvalGoalTime

- EvalGoalType
- EvalGoalStatus
- EvalGoalWhoType

EvalGoalClassifierItem

The EvalGoalClassifierItem table contains information about the progress of the evaluation goal.

Column Name	Description	Storage
id	Numeric identifier for this evaluation form question. This number is auto-generated by the database.	int identity NOT NULL Primary Key
evalGoalFK	A foreign key to the ID in the EvalGoal table.	int NOT NULL
ordinal	The order number for this item.	tinyint NOT NULL
name	The name of this item.	varchar[256] NOT NULL
classifierTypeFK	A foreign key to the ID in the EvalGoalClassifierType table.	int NOT NULL
field		nvarchar(5012) NOT NULL
fieldKey		nvarchar(128) NOT NULL
lastModified	The last time the EvalGoalClassifierItem was modified.	datetime (16,3) NOT NULL

Every EvalGoalClassifierItem record is associated with the following records:

- EvalGoal
- EvalGoalClassifierType

EvalGoalClassifierType

The ACD table contains information about the evaluation goal classifier type.

Column Name	Description	Storage
id	Numeric identifier for the EvalGoalClassifierType, auto-generated by the database.	tinyint NOT NULL Primary Key
name	Identifies the name of the evaluation goal classifier type.	varchar[64] NOT NULL
lastModified	The last time the evaluation goal classifier type was modified.	datetime (16,3) NOT NULL

Every EvalGoalClassifierTyperecord is associated with the following record:

- EvalGoalClassifierItem

EvalGoalEvaluatorSet

The EvalGoalEvaluatorSet table contains information about the evaluation goal for an evaluator

Column Name	Description	Storage
evalGoalFK	A foreign key to the ID in the EvalGoal table.	int NOT NULL
personFK	A foreign key to the ID in the Person table.	int NOT NULL
activated	The GMT date and time when the evaluation goal was activated for the evaluator.	datetime (16,3) NOT NULL
deactivated	The GMT date and time when the evaluation goal was deactivated for the evaluator.	datetime (16,3) NOT NULL
lastModified	The last time the evaluation goal for an evaluator was modified.	datetime (16,3) NOT NULL

Every EvalGoalEvaluatorSet record is associated with the following records:

- EvalGoal
- Person

EvalGoalProgress

The EvalGoalProgress table contains information about the progress of the evaluation goal.

Column Name	Description	Storage
id	Numeric identifier for this evaluation form question. This number is auto-generated by the database.	int identity NOT NULL Primary Key
evalGoalFK	A foreign key to the ID in the evalGoal table.	int NOT NULL
personFK	A foreign key to the ID in the Person table.	int NOT NULL
ccrFK	A foreign key to the ID in the Ccr table.	bigint NOT NULL
completedTime	The time when the evaluation goal was completed	datetime (16,3) NOT NULL
lastModified	The last time the EvalFormQuestion was modified.	datetime (16,3) NOT NULL

Every EvalGoalProgress record is associated with the following records:

- Ccr
- EvalGoal
- Person

EvalGoalStatus

The EvalGoalStatus table contains information about the status of the evaluation goal.

Column Name	Description	Storage
id	Numeric identifier for the evaluation goal status, auto-generated by the database.	tinyint NOT NULL Primary Key
name	Identifies the name of the evaluation goal status.	varchar[64] NOT NULL
lastModified	The last time the evaluation goal status was modified.	datetime (16,3) NOT NULL

Every EvalGoalStatus record is associated with the following record:

- EvalGoal

EvalGoalTime

The EvalGoalTime table contains information about when the evaluation goal was last modified.

Column Name	Description	Storage
id	Numeric identifier for the evaluation goal time, auto-generated by the database.	tinyint NOT NULL Primary Key
name	Identifies the name of the evaluation goal time.	varchar[64] NOT NULL
lastModified	The last time the evaluation goal time was modified.	datetime (16,3) NOT NULL

Every EvalGoalTime record is associated with the following record:

- EvalGoal

EvalGoalType

The EvalGoalType table contains information about the type of evaluation goal.

Column Name	Description	Storage
id	Numeric identifier for the evaluation goal type, auto-generated by the database.	tinyint NOT NULL Primary Key
name	Identifies the name of the evaluation goal type.	varchar[64] NOT NULL
lastModified	The last time the evaluation goal type was modified.	datetime (16,3) NOT NULL

Every EvalGoalType record is associated with the following records:

- EvalGoal

EvalGoalWhoType

The EvalGoalWhoType table contains information about the type of evaluation goal.

Column Name	Description	Storage
id	Numeric identifier for the evaluation goal who type, auto-generated by the database.	tinyint NOT NULL Primary Key
name	Identifies the name of the evaluation goal who type.	varchar[64] NOT NULL
lastModified	The last time the who type for the evaluation goal was modified.	datetime (16,3) NOT NULL

Every EvalGoalWhoType record is associated with the following record:

EvalGoal

EvalQuestion

Quality Management adds a record to the EvalQuestion table each time a question is scored for a particular evaluation of a recording.

Column Name	Description	Storage
id	Numeric identifier for this evaluation question.	int identity NOT NULL Primary Key
formQuestionFK	A foreign key to the ID in the EvalFormQuestion table.	int NOT NULL
evalFK	A foreign key to the ID in the Eval table.	int NOT NULL
selectedOptionFK	A foreign key to the ID in the EvalFormQuestionOption table.	int NOT NULL
lastModified	The last time the EvalQuestion was modified.	datetime (16,3) NOT NULL

Every EvalQuestion record is associated with the following records:

- Eval
- EvalFormQuestion
- EvalFormQuestionOption

EvalState

Quality Management loads the EvalState table with default evaluation state values after the database schema is created. Each record in this table is associated with an evaluation state for a Ccr record. The defaults values are shown in the following table.

ID	Name
0	Unscored
1	Scored
2	In-Progress
3	Needs Approval

The EvalState table contains the fields shown in the following table.

Column Name	Description	Storage
id	Numeric identifier for this evaluation state.	tinyint NOT NULL Primary Key
name	Name of this evaluation state.	varchar(32) NOT NULL

Every EvalState record is associated with the following records:

- Ccr
- Eval

Evaluator

The Evaluator table contains information about specific evaluators. A record is created when an evaluator role is assigned in Quality Management Administrator. After saved in Quality Management Administrator, it will be at most ten minutes until the Monitoring and Recording Sync service adds this record to the database. Activation and deactivation dates reflect the time of the database sync that created or deactivated the record.

Column Name	Description	Storage
personFK	A foreign key to the ID in the Person table.	int NOT NULL
teamFK	A foreign key to the ID in the Team table.	int NOT NULL

Column Name	Description	Storage
activated	The GMT date and time when the agent or knowledge worker role was assigned to the person.	datetime (16,3) NOT NULL
deactivated	The GMT date and time when the agent or knowledge worker role was removed from the person.	datetime (16,3) NOT NULL

Every Agent record is associated with the following records:

- Person
- Team

EvaluatorType

The EvaluatorType table contains information on the types of administrator roles.

Column Name	Description	Storage
id	Numeric identifier for the evaluator role type, auto-generated by the database.	int identity NOT NULL Primary Key
name	The value used for localization.	varchar[32] NOT NULL
text	The English translation of the name column.	varchar[64] NOT NULL

Every EvaluatorType record is associated with the following record:

- Person

EventLogging

The EventLogging table contains information on the event logging.

Column Name	Description	Storage
id	Numeric identifier for the logging event, auto-generated by the database.	bigint identity NOT NULL Primary Key

Column Name	Description	Storage
eventLoggingTypeFK	A foreign key to the ID in the eventLoggingType table.	int NOT NULL
serverFK	A foreign key to the ID in the server table.	int NOT NULL
eventDateTime	The GMT date and time when the event occurred.	datetime (16,3) NOT NULL

Every EventLogging record is associated with the following records:

- EventLoggingType
- EventLoggingParameter
- Server

EventLoggingParameter

The EventLoggingParameter table contains information on the event logging parameter.

Column Name	Description	Storage
id	Numeric identifier for the logging event parameter, auto-generated by the database.	bigint identity NOT NULL Primary Key
eventLoggingFK	A foreign key to the ID in the eventLogging table.	int NOT NULL
name	Identifies the name of the event logging parameter.	varchar[20] NOT NULL
value	The value for the event logging parameter.	nvarchar[500] NOT NULL
listOrder		int NOT NULL

Every EventLoggingParameter record is associated with the following record:

- EventLoggingType

EventLoggingType

The ACD table contains information about the event logging type.

Column Name	Description	Storage
id	Numeric identifier for the event logging type, auto-generated by the database.	int identity NOT NULL Primary Key
name	Identifies the name of the event logging type.	varchar[256] NOT NULL

Every EventLoggingType record is associated with the following record:

- EventLogging

Field

Quality Management loads the Field table with the valid fields.

Column Name	Description	Storage
id	Numeric identifier for this field. This number is auto-generated by the database.	int identity NOT NULL Primary Key
name	Name of this field. **Example:** Group Name or Date.	varchar(50) NOT NULL
isEnabled	0—Display this field in Unified Workforce Optimization. 1—Do not display this field in Unified Workforce Optimization.	bit NOT NULL
categoryFK	A foreign key to the ID in the FieldCategory table.	int NOT NULL
featureFK	A foreign key to the id in the UI_Feature table.	int NOT NULL

Every Field record is associated with the following records:

- FieldCategory
- PersonField
- UI_Feature

FieldCategory

Quality Management loads the FieldCategory table with the valid field categories.

Column Name	Description	Storage
id	Numeric identifier for this field category. This number is auto-generated by the database.	int NOT NULL Primary Key
name	Name of this field category. **Example:** Organization Fields or Evaluation Fields.	varchar(50) NOT NULL

Every FieldCategory record is associated with the following record:

- Field

FileType

Quality Management loads the FileType table with default file type values after the database schema is created. The default values are shown in the following table.

ID	Extension	Name	isAudio	isScreen
2	rec	Funk Screen Recording	0	1
3	spx	Speex Audio Recording	1	0

The FileType table contains the fields shown in the following table.

Column Name	Description	Storage
id	Numeric identifier for this file type.	tinyint NOT NULL
extension	File type extension.	varchar(16) NOT NULL
name	Name of this file type.	varchar(64) NOT NULL
isAudio	1—an audio file type 0—not an audio file type	bit NOT NULL

Column Name	Description	Storage
isScreen	1–a screen file type 0–not a screen file type	bit NOT NULL
isPlayable	1–the file is playable 0–the file is not playable	bit NOT NULL
mimeType	The recording's file type format.	varchar(32) NOT NULL

Every FileType record is associated with the following record:

- MediaFile

Filter

Quality Management loads the Filter table with the valid searches.

Column Name	Description	Storage
id	Numeric identifier for this search. This number is auto-generated by the database.	int identity NOT NULL Primary Key
name	Name of this search. **Example:** Search Recordings or Search Reporting	nvarchar(128) NOT NULL
value	The selection criteria for the search.	nvarchar (1073741823) NOT NULL
personFK	A foreign key to the id in the Person table.	int NOT NULL
featureFK	A foreign key to the id in the UI_Feature table.	int NOT NULL

Every Filter record is associated with the following records:

- Person
- UI_Feature

FilterParameter

Quality Management loads the search filter parameters.

Column Name	Description	Storage
id	Numeric identifier for this filter parameter. This number is auto-generated by the database.	int identity NOT NULL Primary Key
name	Name of this filter parameter.	varchar(255) NOT NULL

Every FilterParameter record is associated with the following records:

- FilterProperties
- RequiredFilterParam

FilterProperties

Quality Management loads the search filter properties.

Column Name	Description	Storage
id	Numeric identifier for this filter parameter. This number is auto-generated by the database.	int identity NOT NULL Primary Key
filterFK	A foreign key to the Filter table.	int NOT NULL
filterParamFK	A foreign key to the FilterParameter table.	int NOT NULL
SettingsValue	The selection criteria for the search filter.	varchar(250) NOT NULL

Every FilterProperties record is associated with the following records:

- SyncFilter
- FilterParameter

GamificationConnectionInfo

Quality Management loads the gamificationConnectionInfo table with the valid connection information for gamification.

Column Name	Description	Storage
id	Numeric identifier for this gamification connection information.	bigint identity NOT NULL Primary Key
serverFK	A foreign key to the ID in the Server table.	bigint NOT NULL
gamificationEventTypeFK	A foreign key to the ID in the gamificationEventType table.	int NOT NULL
fetchCommand	The path to the metric data on the server.	varchar[1500] NOT NULL
cronSchedule	A CRON schedule string.	varchar[1500] NOT NULL

Every gamificationConnectionInfo record is associated with the following records:

- Server
- GamificationEventType

GamificationEvent

GamificationEvent contains the records of events for gamification.

Column Name	Description	Storage
id	Numeric identifier for this gamification event.	bigint identity NOT NULL Primary Key
personFK	A foreign key to the ID in the Person table.	bigint NOT NULL
gamificationEventTypeFK	A foreign key to the ID in the gamificationEventType table.	int NOT NULL

Column Name	Description	Storage
rawScore	The raw score for the event.	float NULL
eventDateTime	The GMT date and time when the event occurred.	datetime (16,3) NOT NULL
insertDateTime	The GMT date and time when the insert occurred.	datetime (16,3) NOT NULL
eventId	Numeric identifier for the event, auto-generated by the database.	bigint identity NOT NULL

Every GamificationEvent record is associated with the following records:

- Person
- GamificationEventType

GamificationEventType

GamificationEventType contains the records of event types for gamification

Column Name	Description	Storage
id	Numeric identifier for this gamification event type.	bigint identity NOT NULL Primary Key
name	Identifies the name of the gamification event type.	varchar[128] NOT NULL
isDataPushed	0–the data is not pushed. 1–the data is pushed.	bit NOT NULL
isEnabled	0–the gamification event type is not enabled. 1–the gamification event type is enabled.	bit NOT NULL

Every GamificationEventType record is associated with the following records:

- GamificationConnectionInfo
- GamificationEvent

- GamificationLevelPointRange

GamificationIcon

GamificationIcon contains the records of icons for gamification.

Column Name	Description	Storage
id	Numeric identifier for this gamification icon.	int identity NOT NULL Primary Key
name	Identifies the name of the gamification icon.	varchar[128] NOT NULL
iconPath	The location where the icon resides.	varchar(3000) NOT NULL

Every GamificationIcon record is associated with the following record:

- GamificationLevelScoring

GamificationLevel

GamificationLevel contains the records of levels for gamification.

Column Name	Description	Storage
id	Numeric identifier for this gamification level.	int identity NOT NULL Primary Key
name	Identifies the name of the gamification level.	varchar[128] NOT NULL
dateRange	A date range for the gamificationLevel	int NOT NULL

Every GamificationLevel record is associated with the following records:

- GamificationLevelPointRange
- GamificationLevelScoring

GamificationLevelPointRange

GamificationLevelPointRange contains the records of the level point ranges for gamification.

Column Name	Description	Storage
id	Numeric identifier for this gamification level.	int identity NOT NULL Primary Key
name	Identifies the name of the gamification level.	varchar[128] NOT NULL
gamificationPointRangeFK	A foreign key to the ID in the gamificationPointRange table.	int NOT NULL
gamificationLevelFK	A foreign key to the ID in the gamificationLevel table.	int NOT NULL
gamificationEventTypeFK	A foreign key to the ID in the gamificationEventType table.	int NOT NULL

Every GamificationLevelPointRange record is associated with the following records:

- GamificationEventType
- GamificationLevel
- GamificationLevelPointRangePerson
- GamificationPointRange

GamificationLevelPointRangePerson

GamificationLevelPointRangePerson contains the records of the level point range person for gamification

Column Name	Description	Storage
id	Numeric identifier for this gamification level.	int identity NOT NULL Primary Key
gamificationLevelPointRangeFK	A foreign key to the ID in the gamificationLevelPointRange table.	int NOT NULL
personFK	A foreign key to the ID in the person table.	int NOT NULL

Every GamificationLevelPointRangePerson record is associated with the following records:

- GamificationLevelPointRange
- Person

GamificationLevelScoring

GamificationLevelScoring contains the records of the level scoring for gamification.

Column Name	Description	Storage
id	Numeric identifier for this gamification level.	int identity NOT NULL Primary Key
minScoreThreshold	The minimum threshold for the score.	float NULL
gamificationIconFK	A foreign key to the ID in the gamificationIcon table.	int NOT NULL
gamificationLevelFK	A foreign key to the ID in the gamificationLevel table.	int NOT NULL

Every GamificationLevelScoring record is associated with the following records:

- GamificationIcon
- GamificationLevel

GamificationPointRange

GamificationPointRange contains the records of point ranges for gamification.

Column Name	Description	Storage
id	Numeric identifier for this gamification level.	int identity NOT NULL Primary Key
name	Identifies the name of the gamification point range.	varchar[128] NOT NULL

Every GamificationPointRange record is associated with the following records:

- GamificationLevelPointRange
- GamificationPointRangeScoring

GamificationPointRangeScoring

GamificationPointRangeScoring contains the records of point range scoring for gamification.

Column Name	Description	Storage
id	Numeric identifier for this gamification point range scoring.	int identity NOT NULL Primary Key
rangeMinimum	The minimum range for the score.	float NULL
gamificationPointRangeFK	A foreign key to the ID in the gamificationPointRange table.	int NOT NULL

Every GamificationPointRangeScoring record is associated with the following records:

- GamificationPointRange

Gateway

The Gateway table contains configuration information for SPAN recording.

Column Name	Description	Storage
id	Numeric identifier for this gateway.	bigint identity NOT NULL Primary Key
monitorServerFK	A foreign key to the ID in the Server table.	bigint NOT NULL
ipHostName	The IP address or hostname entered by the administrator in Quality Management Administrator when configuring gateways for a remote agent for Server Recording (SPAN).	varchar[40] NOT NULL

Every Gateway record is associated with the following record:

- Server

GetRandNumber

The GetRandNumber view generates a random number from SQL. This random number is used by Workflow to randomly select recordings.

Column Name	Description	Storage
num	A random number of type uniqueidentifier from SQL.	uniqueidentifier NOT NULL

Groups

A record is added to the Groups table every time a group is created and saved in Quality Management Administrator.

After being saved in Quality Management Administrator, it will be at most ten minutes until the Monitoring and Recording Sync service adds this record to the database. Activation and deactivation dates reflect the time of the database sync that created or deactivated the record.

Column Name	Description	Storage
id	Numeric identifier for this group. This number is auto-generated by the database.	int identity NOT NULL Primary Key
name	Name of this group.	nvarchar(64) NOT NULL
activated	The GMT date and time when the group was created.	datetime (16,3) NOT NULL
deactivated	The GMT date and time when the group was removed.	datetime (16,3) NOT NULL

Every Groups record is associated with the following records:

- Ccr
- Manager
- TeamGroup

LicenseBundle

The LicenseBundle table contains information about the license bundle for Quality Management.

Column Name	Description	Storage
id	Numeric identifier for the license bundle, auto-generated by the database.	int identity NOT NULL Primary Key
name	Identifies the name of the license bundle.	varchar[32] NOT NULL

Every LicenseBundle record is associated with the following record:

- Ccr

LicenseLibrary

The LicenseLibrary table contains license information. The data in the table is encrypted. Cisco does not publish the encryption key.

Column Name	Description	Storage
id	Numeric identifier for the license library.	varchar(32) NOT NULL
data	Contains license information in encrypted format.	varchar (2147483647) NOT NULL

LoginState

The LoginState table contains the ACD and Quality Management login states.

Column Name	Description	Storage
id	The unique ID for the state.	int identity NOT NULL Primary Key
personFK	The foreign key to the Person table.	int NOT NULL
realmFK	Indicates if this is a Monitoring and Recording state change or an ACD state change.	tinyint NOT NULL

Column Name	Description	Storage
extension	The extension(s) where the login/logout occurred.	varchar (2147483647) NOT NULL
login	The date and time of the login.	datetime (16,3)
logout	The date and time of the logout.	datetime (16,3)
versionInfo	The version of the Desktop Recording service.	varchar (25) NOT NULL

Every LoginState record is associated with the following records:

- Person

Manager

Quality Management creates a record in the Manager table when a group is assigned to a manager.

After being saved in Quality Management Administrator, it will be at most ten minutes until the Monitoring and Recording Sync service adds this record to the database. Activation and deactivation dates reflect the time of the database sync that created or deactivated the record.

Column Name	Description	Storage
personFK	A foreign key to the ID in the Person table. This is the person managing the group.	int NOT NULL
groupFK	A foreign key to the ID in the Groups table. This is the group being managed.	int NOT NULL
activated	The GMT date and time when the group was assigned to the manager.	datetime (16,3) NOT NULL
deactivated	The GMT date and time when the group was removed from the manager.	datetime (16,3) NOT NULL

Every Manager record is associated with the following records:

- Person
- Groups

Media

Quality Management creates a new record in the Media table every time an audio or screen recording is uploaded to the Quality Management Record Server.

Column Name	Description	Storage
id	Numeric identifier for this audio or record, auto-generated by the database.	bigint identity NOT NULL Primary Key
icmCallId	The Peripheral Call Key (or Call ID) for the call from the Unified CM system.	varchar(256) NOT NULL
isReconciled	0—the audio or screen recording is reconciled. 1—the audio or screen recording is not reconciled.	bit NOT NULL
startTime	The start time for the event.	datetime (16,3) NOT NULL
duration	The duration of the recording in milliseconds.	int NOT NULL
offset	The time the recording starts relative to the startTime timestamp of the Ccr, in milliseconds. A positive value indicates the recording starts after the startTime and negative value indicates that it starts before.	int NOT NULL
mediaTypeFK	A foreign key to the ID in the MediaType table. **Example:** rec	tinyint NOT NULL
personFK	The foreign key to the Person table.	int NOT NULL
isDeletePending	0—indicates the media is not pending for deletion by the Database Cleaner Service. 1—indicates that the media is pending for deletion.	bit NOT NULL

Every Screen record is associated with the following records:

- Ccr
- MediaFile
- MediaType

MediaFile

MediaFile contains records of audio and screen media files.

Column Name	Description	Storage
id	Numeric identifier for the MediaFile, auto-generated by the database.	bigint NOT NULL Primary Key
mediaFK	A foreign key to the ID in the Media table.	bigint NOT NULL
pathFK	A foreign key to the ID in the Path table.	int NOT NULL
fileTypeFK	A foreign key to the ID in the FileType table. **Example:** `spx`	tinyint NOT NULL
serverFK	A foreign key to the ID in the Server table.	bigint NOT NULL
fileSize	The size of the audio recording.	bigint NOT NULL
displayName	The display name for this media file.	nvarchar(20) NOT NULL
resolution	The resolution setting for this media file.	nvarchar(30) NOT NULL
coordinates	The coordinates for this media file.	nvarchar(30) NOT NULL
uploadStateFK	A foreign key to the ID in the UploadState table.	tinyint NOT NULL
fileName	The file name for this media file.	nvarchar(510) NOT NULL

Every MediaFile record is associated with the following records:

- Media
- Path
- FileType
- Server
- UploadState

MediaType

The MediaType table contains information on the types of media.

Column Name	Description	Storage
id	Numeric identifier for the media type, auto-generated by the database.	int identity NOT NULL Primary Key
name	Identifies the name of the media type.	varchar[25] NOT NULL

Every MediaType record is associated with the following record:

- Media

MetaData

A MetaData record is created or updated when the metadata for a Ccr is inserted or edited from the Quality Management application in Unified Workforce Optimization or the Recording API for Quality Management.

Column Name	Description	Storage
ccrFK	A foreign key to the ID in the Ccr table.	bigint NOT NULL
metaDataFieldFK	A foreign key to the ID in the MetaDataField table.	int NOT NULL
data	The metadata (encrypted if the metadata field is set to isEncrypted = 1).	nvarchar(4112) NOT NULL
hashedData	Save data in hash format.	varbinary NOT NULL

Column Name	Description	Storage
lastModified	The last time the metadata was modified.	datetime (16,3) NOT NULL

Every MetaData record is associated with the following records:

- MetaDataField
- Ccr

MetaDataField

A new record is added to the MetaDataField table for every user-defined metadata field that is configured in Quality Management Administrator.

Column Name	Description	Storage
id	Numeric identifier for this metadata field. This number is auto-generated by the database.	int identity NOT NULL Primary Key
displayName	The display name of the metadata field. This is displayed to the user in Unified Workforce Optimization.	nvarchar(128) NOT NULL
keyName	The unique key name of the metadata field that cannot be edited after creation of the metadata field.	nvarchar(78) NOT NULL
typeFK	A foreign key to the ID in the MetaDataType table.	smallint NOT NULL
isEncrypted	0—if the metadata for this metadata field will not be encrypted in the database. 1—if the metadata for this metadata field will be encrypted.	bit NOT NULL
isExportable	0—if the metadata for this metadata field will not be added as metadata to exported media files. 1—if the metadata for this metadata field will be added as metadata to exported media files.	bit NOT NULL
isReadOnly	0—if the this metadata field is not read-only. 1—if this metadata field is read-only. Read-only fields cannot be created within Quality Management Administrator.	bit NOT NULL

Column Name	Description	Storage
displayOrder	The display order for this metadata field	int NOT NULL
mappedColumnFK	A foreign key to the ID in the MetaDataMappableColumns table.	bigint NOT NULL
lastModified	The last time the metadata field was modified.	datetime (16,3) NOT NULL

Every MetaDataField record is associated with the following records:

- MetaData
- MetaDataType
- MetaDataMappableColumns
- RecordingApiCommand

MetaDataMappableColumns

A new record is added to the metaDataMappableColumns table for every mappable metadata column that is configured in Quality Management Administrator.

Column Name	Description	Storage
id	Numeric identifier for this metadata field. This number is auto-generated by the database.	bigint identity NOT NULL Primary Key
mappedColumn	The name of the mapped column	nvarchar(256) NOT NULL
acdType	The ACD type.	smallint NOT NULL

Every metaDataMappableColumns record is associated with the following records:

- MetaDataField

MetaDataType

Quality Management loads the MetaDataType table with default metadata type values after the database schema is created. The default values for the type of user-defined metadata are shown in the following figure.

ID	Name
0	Text
1	Number
2	Date

The MetaDataType table contains the fields shown in the following table.

Column Name	Description	Storage
id	Numeric identifier for this MetaDataType.	smallint NOT NULL Primary Key
name	Name of this MetaDataType.	nvarchar(64) NOT NULL
lastModified	The last time the metadata type was modified.	datetime (16,3) NOT NULL

Every MetaDataType record is associated with the following record:

- MetaDataField

Path

A record is inserted in the Path table any time a recording is uploaded to a new path. The path name indicates the relative path to which a recording was uploaded. The full path for retrieving this recording is constructed by appending this to the screenBasePath or audioBasePath of the host. The forward slash '/' character must be used as the separator for multiple directory names.

Column Name	Description	Storage
id	Numeric identifier for this path. This number is auto-generated by the database.	int identity NOT NULL Primary Key
name	Name of this path. The realm.skillTargetId of the person is used for the name.	nvarchar(510) NOT NULL

Every Path record is associated with the following records:

- MediaFile

Person

A record is inserted in the Person table any time a person is configured and saved in Quality Management Administrator.

Column Name	Description	Storage
id	Numeric identifier for this person. This number is auto-generated by the database.	int identity NOT NULL Primary Key
lastName	Last name of this person.	nvarchar(120) NOT NULL
firstName	First name of this person.	nvarchar(120) NOT NULL
username	The windows login of the person. This is entered in Quality Management Administrator in Quality Management authentication or comes from Active Directory in AD authentication.	nvarchar(128) NOT NULL
domainName	The domain that the person belongs to for AD authentication. For Quality Management authentication this will be an empty string.	nvarchar(128) NOT NULL
locale	The locale of the person.	varchar(16) NOT NULL
realmFK	A foreign key to the ID in the Realm table.	tinyint NOT NULL
skillTargetId	Realm.skillTargetId is the unique identifier used for a person. The skillTargetId is the unique ID retrieved from the ACD database for agents or auto-generated for knowledge workers in Quality Management.	nvarchar(120) NOT NULL
isEvaluator	0—if the person has the evaluator role. 1—if the person does not have the evaluator role.	bit NOT NULL
activated	The GMT date and time when the person was created (in the case of knowledge worker) or configured (from the ACD).	datetime (16,3) NOT NULL

Column Name	Description	Storage
deactivated	The GMT date and time when the person was deleted.	datetime (16,3) NOT NULL
displayId	The agent identifier displayed in Quality Management Administrator. For more information, see Dates in the Database.	nvarchar(120) NOT NULL
isArchiveUser	0–if the person has the archive user role. 1–if the person does not have the archive user role.	bit NOT NULL
isConfigured	0–if the person is configured for Quality Management. 1–if the person is not configured for Quality Management.	bit NOT NULL
isManager	0–if the person has the manager role. 1–if the person does not have the manager role.	bit NOT NULL
isSupervisor	0–if the person has the supervisor role. 1–if the person does not have the supervisor role.	bit NOT NULL
password	The user’s password.	varchar[128] NOT NULL
acdFirstName	The first name of an ACD user.	nvarchar[120] NOT NULL
acdLastName	The last name of an ACD user.	nvarchar[120] NOT NULL
crsResoureId	The ACD’s ID.	nvarchar[120] NOT NULL
isHotdeskDefaultUser	0–if the person is the default Hot Desk user. 1–if the person is not the default Hot Desk user.	bit NOT NULL

Column Name	Description	Storage
isAdministrator	0–if the person has the administrator user role. 1–if the person does not have the administrator user role.	bit NOT NULL
isBusinessAdministrator	0–if the person has the business administrator user role. 1–if the person does not have the business administrator user role.	bit NOT NULL
isTelephony Administrator	0–if the person has the telephony administrator user role. 1–if the person does not have the telephony administrator user role.	bit NOT NULL
isSystemAdministrator	0–if the person has the system administrator user role. 1–if the person does not have the system administrator user role.	bit NOT NULL
evaluatorTypeFK	A foreign key to the ID in the evaluatorType table.	int NOT NULL
isAutoAssignedEvaluations	0–if the person is automatically assigned evaluations. 1–if the person is not automatically assigned evaluations.	bit NOT NULL
lastModified	The last time the person was modified.	datetime (16,3) NOT NULL
acdFK	A foreign key to the ID in the ACD table.	int NOT NULL

Every Person record is associated with the following records:

- ACD
- Agent
- Alert
- ArchiveAudit

- Ccr
- DashUserView
- DashWidgetViewSetting
- Eval
- EvalComment
- EvalGoalEvaluatorSet
- EvalGoalProgress
- Evaluator
- EvaluatorType
- Filter
- GamificationEvent
- GamificationLevelPointRangePerson
- LoginState
- Manager
- PersonField
- Realm
- QMAdminAudit
- RealTimeRecordingMonitorState
- RecordingApiCommand
- RecordingStateAudit
- ReportUserConfig
- UserReport
- ScreenMonitoring
- VoiPMonitorDevice
- WorkflowRuleWhoPerson

PersonField

Quality Management loads the PersonField table with the valid fields for each person.

Column Name	Description	Storage
fieldFK	A foreign key to the ID in the Field table.	int NOT NULL
personFK	A foreign key to the ID in the Person table.	int NOT NULL
isSelected	0—if the person is selected. 1—if the person is not selected.	bit NOT NULL
ordinal	The order number for this PersonField.	int NOT NULL
width	The width of the Person field that appears in the Recordings, Live Monitoring, and Recording Monitoring applications.	int NOT NULL

Every PersonField record is associated with the following records:

- Field
- Person

QMAdminAudit

Quality Management loads the QMAdminAudit table with the valid QM administrator audit information.

Column Name	Description	Storage
id	Numeric identifier for the QM Admin Audit, auto-generated by the database.	int identity NOT NULL Primary Key
timestamp	The date and time of the QM Admin Audit command occurred.	datetime (16,3)
personFK	A foreign key to the ID in the Person table.	int NOT NULL
actionFK	A foreign key to the ID in the QMAdminAuditAction table.	int NOT NULL
areaFK	A foreign key to the ID in the QMAdminAuditArea table.	int NOT NULL

Column Name	Description	Storage
jsonDiff		nvarchar[4112]
stringDiff		nvarchar[4112]

Every QMAdminAudit record is associated with the following records:

- Person
- QMAdminAuditAction
- QMAdminAuditArea

QMAdminAuditAction

Quality Management loads the QMAdminAuditAction table with the valid QM administrator audit action information.

Column Name	Description	Storage
id	Numeric identifier for this QM admin audit action.	int NOT NULL Primary Key
name	Identifies the name of the QM admin audit action.	varchar[32] NOT NULL

Every AlertType record is associated with the following record:

- QMAdminAudit

QMAdminAuditArea

Quality Management loads the QMAdminAuditArea table with the valid QM administrator audit area information.

Column Name	Description	Storage
id	Numeric identifier for this QM admin audit area.	int NOT NULL Primary Key
name	Identifies the name of the QM admin audit area.	varchar[128] NOT NULL

Every AlertType record is associated with the following record:

- QMAdminAudit

Realm

Quality Management loads the Realm table with the default Realm values after the database schema is created. The Realm table is used as a foreign key in other tables to indicate whether a team was created in Quality Management Administrator or synced into Quality Management Administrator from the ACD.

The defaults values are shown in the following table.

ID	Name
0	Quality Management
1	ICM

The Realm table contains the fields shown in the following table.

Column Name	Description	Storage
id	Numeric identifier for this realm.	tinyint NOT NULL Primary Key
name	Name of this realm.	varchar(32) NOT NULL

Every Realm record is associated with the following record:

- Person

RealTimeRecordingMonitorState

The RealTimeRecordingMonitorState table specifies the current recording state.

Column Name	Description	Storage
id	The unique ID for the state.	int identity NOT NULL Primary Key
personFK	Foreign key to the Person table.	int NOT NULL

Column Name	Description	Storage
recordingStateFK	Foreign key to the RecordingStateCause table.	int NOT NULL
recordingEventType	The failure cause.	varchar (2147483647) NOT NULL
failureTime	The date and time of the last failure.	datetime (16,3)
successTime	The date and time of the last success.	datetime (16,3)

Every RealTimeRecordingMonitorState record is associated with the following records:

- Person
- RecordingStateCause

ReconciliationHistory

A record is inserted in the ReconciliationHistory table each time a call is reconciled.

Column Name	Description	Storage
id	Numeric identifier for this reconciliation. This number is auto-generated by the database.	int identity NOT NULL Primary Key
cdrIdFK	A foreign key to the Cdr table.	varchar(100) NOT NULL
ccrIdFK	A foreign key to the pkid in the Ccr table.	bigint NOT NULL
callLegReason		smallint NOT NULL
personMatchStatus		smallint NOT NULL
datetimeUTC	The GMT (UTC) date and time when the call was reconciled.	datetime (16,3) NOT NULL
acdFK	A foreign key to the pkid in the ACD table.	int NOT NULL

Every ReconciliationHistory record is associated with the following records:

- ACD
- Ccr

ReconciliationHistoryStatus

The ReconciliationHistoryStatus table contains information about the Automatic Call Distributor (ACD).

Column Name	Description	Storage
id	Numeric identifier for the reconciliation history status, auto-generated by the database.	int identity NOT NULL Primary Key
description	A description of the reconciliation history status.	varchar[256] NOT NULL

RecordingApiCommand

The RecordingApiCommand table specifies the Recording API command.

Column Name	Description	Storage
id	The unique ID for the Recording API command.	int identity NOT NULL Primary Key
typeFK	Foreign key to the RecordingApiCommandType table.	int NOT NULL
personFK	Foreign key to the Person table.	int NOT NULL
timestamp	The date and time of the Recording API command was issued.	datetime (16,3)
metaDataFieldFK	Foreign key to the metaDataField table.	int NOT NULL
data	The failure cause.	varchar (4112) NOT NULL

Column Name	Description	Storage
activeCallOnly	0—the recording API command is not marked for active call only. 1—the recording API command is marked for active call only.	bit NOT NULL

Every RecordingApiCommand record is associated with the following records:

- RecordingApiCommandType
- Person
- MetaDataField

RecordingApiCommandType

The RecordingApiCommandType table tracks the Recording API commands sent by users and includes data sent if applicable.

Column Name	Description	Storage
id	The unique ID for the Recording API command type.	int identity NOT NULL Primary Key
Name	The name of this Recording API command type.	varchar(32) NOT NULL

Every RealTimeRecordingMonitorState record is associated with the following records:

- RecordingApiCommand

RecordingCluster

A record is inserted in the RecordingCluster table any time a recording cluster is configured and saved in Quality Management Administrator.

Column Name	Description	Storage
id	Numeric identifier for this recording cluster. This number is auto-generated by the database.	int identity NOT NULL Primary Key
Name	The name of this recording cluster.	varchar(64) NOT NULL

Column Name	Description	Storage
siteFK	A foreign key to the ID in the Site table.	int NOT NULL
signalingGroupFK	A foreign key to the ID in the SignalingGroup table.	int NOT NULL

Every RecordingCluster record is associated with the following records:

- RecordingClusterServer
- SignalingGroup
- Site
- VoiPMonitorDevice

RecordingClusterServer

A record is inserted in the RecordingClusterServer table any time a Record Server is configured for a recording cluster and saved in Quality Management Administrator.

Column Name	Description	Storage
recordingClusterFK	A foreign key to the ID in the RecordingCluster table.	int NOT NULL
serverFK	A foreign key to the ID in the Server table.	bigint NOT NULL
priority	The priority for the server in the recording cluster.	tinyint NULL

Every RecordingClusterServer record is associated with the following records:

- RecordingCluster
- Server

RecordingEvent

The RecordingEvent table contains records of recording events.

Column Name	Description	Storage
id	Numeric identifier for this recording event. This number is auto-generated by the database.	bigint identity NOT NULL Primary Key
ccrFK	A foreign key to the ID in the Ccr table.	bigint NOT NULL Primary Key
recordingEventTypeFK	A foreign key to the ID in the RecordingEventType table.	smallint NOT NULL
audioRecordingOffset	The offset (in milliseconds) from the start of the audio recording.	int NOT NULL
duration	The duration of the recording event in milliseconds.	int NOT NULL
text	A longer description of this recording event.	nvarchar(512)
audioEditRequired	0–the audio does not require editing. 1–the audio requires editing.	bit NOT NULL
screenEditRequired	0–the screen does not require editing. 1–the screen requires editing.	bit NOT NULL
lastModified	The last time the recording event was modified.	datetime (16,3) NOT NULL

Every RecordingEvent record is associated with the following records:

- Ccr
- RecordingEventType

RecordingEventType

The RecordingEventType table contains a record for each type of recording event that Quality Management supports (for example, talkover and silence).

Column Name	Description	Storage
name	Name of this recording event type.	varchar(64) NOT NULL Primary
categoryFK	A foreign key to the ID in the RecordingEventTypeCategory table.	smallint NOT NULL
id	Numeric identifier for this recording event.	smallint NOT NULL Primary Key
displayName	The display name for this recording event type.	nvarchar(256)
lastModified	The last time the recording event type was modified.	datetime (16,3) NOT NULL

Every RecordingEventType record is associated with the following record:

- RecordingEventTypeCategory
- RecordingEvent

RecordingEventTypeCategory

The RecordingEventTypeCategory table contains a record of the recording event type's category.

Column Name	Description	Storage
id	Numeric identifier for this recording event type category.	smallint NOT NULL Primary Key
name	Name of this recording event type.	varchar(64) NOT NULL

Every RecordingEventTypeCategory record is associated with the following record:

- RecordingEventType

RecordingReason

The RecordingReason table contains a record for each reason why a Ccr is or is not recorded for quality or archive purposes. Quality Management loads the RecordingReason table with the default RecordingReason values after the database schema is created. Negative values indicate that the Ccr was not recorded and positive values indicate that it was recorded. The name field contains keys to externalized strings. The localized values appear in Unified Workforce Optimization.

The default values are shown in the following table.

ID	Name	Text	English String value for Name key
–15	rec_reason_do_not_record	Classifier is do not record	Do not record
–14	rec_reason_no_rules_match	Call does not match any Rule for the Workflow	No rule match
–13	rec_reason_no_recording	No Recording	No Recording
–12	rec_reason_no_sip_event	No SIP event	No SIP event
–11	rec_reason_compliance_license	Agent Licensed for Compliance Only	Agent Licensed for Compliance Only
–10	rec_reason_archive_off	Archive Off	Archive Off
–9	rec_reason_no_rules	There Are No Rules Defined for the Classifier	There Are No Rules Defined for the Classifier
–8	rec_reason_not_who	Call Does Not Match the Who Rule for the Workflow	Not WF Who
–7	rec_reason_not_when	Call Does Not Match the When Rule for the Workflow	Not WF When
–6	rec_reason_not_wf_classifier	Not Meeting Workflow Classifier Calling/Called Number	Not WF Classifier
–5	rec_reason_under_min	Under the Minimum Duration	Under Min duration
–4	rec_reason_in_exclusion	Extension In Exclusion List	In Exclusion
–3	rec_reason_not_inclusion	Extension Not in Inclusion List	Not in Inclusion
–2	rec_reason_deleted_client	Deleted	Delete

ID	Name	Text	English String value for Name key
–1	rec_reason_no_wf_team	No Workflow Configured for Agent's Team	No WF for Team
0	rec_reason_not_processed	No Processed	None
1	rec_reason_what_first	First Call of Day	First
2	rec_reason_what_last	Last Call of Day	Last
3	rec_reason_what_longest	Longest Call of Day	Longest
4	rec_reason_what_shortest	Shortest Call of Day	Shortest
5	rec_reason_what_random_1	Random Call	Random Call 1
6	rec_reason_what_new_emp	New Employee	New Employee
7	rec_reason_what_performance	Performance	Performance
8	rec_reason_what_logging	Logging	Logging
9	rec_reason_tagged	Agent Tagged	Tagged
10	rec_reason_archive	Archive On	Archive
11	rec_reason_analytics_tagged	Analytics Tagged	Analytics Tagged
12	rec_reason_mark_quality	Marked For Quality	Marked For Quality
13	rec_reason_noncall	Non-Call Evaluation	Non-Call Evaluation
14	rec_reason_tagged_archive	Archive Tagged	Archive Tagged
15	rec_reason_api_imported	API Imported	API Imported
16	rec_reason_survey_kpi	Survey Results Triggered	Survey Results Triggered
17	rec_reason_untagged	Untagged	Untagged

ID	Name	Text	English String value for Name key
18	rec_reason_untagged_archive	Archive Untagged	Archive Untagged
19	rec_reason_what_random_2	Random Call 2	Random Call 2
20	rec_reason_what_random_3	Random Call 3	Random Call 3
21	rec_reason_what_random_4	Random Call 4	Random Call 4
22	rec_reason_what_random_5	Random Call 5	Random Call 5

The RecordingReason table contains the fields shown in the following table.

Column Name	Description	Storage
id	Numeric identifier for this recording reason.	smallint NOT NULL Primary Key
name	Name of this recording reason. This field contains keys for strings externalized for localization.	varchar(32) NOT NULL
text	Longer description of this recording reason.	varchar(64) NOT NULL

Every RecordingReason record is associated with the following records:

- Ccr—Every Ccr record has two associated records in the RecordingReason table, one indicates the reason why the Ccr was or was not recorded for Archiving and the other indicates the reason why the Ccr was or was not recorded for Quality Management purposes.
- WorkflowRuleWhat

RecordingStateAudit

The RecordingStateAudit table contains the audit information for the recording state.

Column Name	Description	Storage
id	The unique ID for the state.	int identity NOT NULL Primary Key
personFK	The foreign key to the Person table.	int NOT NULL
recordingStateCauseFK	The foreign key to the RecordingStateCause table.	int NOT NULL
eventTime	The date and time of the event.	datetime (16,3) NOT NULL
deviceName	The recording device where the event occurred.	nvarchar (510) NOT NULL
recordingIPAddress	The IP address where the recording occurred.	varchar (40) NOT NULL

Every RecordingStateAudit record is associated with the following records:

- Person
- RecordingStateCause

RecordingStateCause

The RecordingStateCause table contains the available failure/success causes for the recording state.

Column Name	Description	Storage
id	The unique for the failure/success cause.	int identity NOT NULL Primary Key
ctiMessage	The message text from CTI.	varchar (2147483647) NOT NULL
displayMessage	The message key used for display.	varchar (2147483647) NOT NULL

Column Name	Description	Storage
stateType	The current state.	varchar (2147483647) NOT NULL
voiceFailure	Indicates if this is a voice failure message. By default this is set to False.	bit NOT NULL
screenFailure	Indicates if this is a screen failure message. By default this is set to False.	bit NOT NULL

Every RecordingStateCause record is associated with the following records:

- RealTimeRecordingMonitorState
- RecordingStateAudit

RecordingType

The RecordingType table indicates the VoIP monitor recording type associated with a VoiPMonitorDevice.

Column Name	Description	Storage
id	Numeric identifier for the recording type. This number is auto-generated by the database.	int NOT NULL Primary Key
name	Name of this recording type. **Example:** Multiple Registration	varchar(25) NOT NULL
isReconciled	0–the recording type is not reconciled. 1–the recording type is reconciled.	bit NOT NULL

Every RecordingType record is associated with the following records:

- Ccr
- VoiPMonitorDevice

Report

The Report table contains information on each report.

Column Name	Description	Storage
id	Numeric identifier for this report.	int identity NOT NULL Primary Key
typeFK	Type of report.	int NOT NULL
name	Name of this report.	varchar(255) NOT NULL

Every Report record is associated with the following records:

- ReportType
- ReportConfiguration
- ReportRequiredParam
- ReportRoleScope
- ReportUserConfig
- UserReport

ReportColumn

The ReportColumn table identifies the columns in a report.

Column Name	Description	Storage
id	Numeric identifier for this file type.	int identity NOT NULL Primary Key
name	Name of this report column.	varchar(255) NOT NULL

Every ReportColumn record is associated with the following records:

- ReportConfiguration
- ReportUserConfig

ReportConfiguration

The ReportConfiguration table contains the report configuration information.

Column Name	Description	Storage
reportFK	A foreign key to the ID in the Report table.	int NOT NULL
reportColumnFK	A foreign key to the ID in the ReportColumn table.	int NOT NULL
isRequired	1–if the report includes this column by default. 0–if the report does not include this column by default.	bit NOT NULL
ordinal	The order number for this report configuration.	tinyint NOT NULL

Every ReportConfiguration record is associated with the following records:

- Report
- ReportColumn

ReportParameter

The ReportParameter table identifies the parameters in a report.

Column Name	Description	Storage
id	Numeric identifier for this file type.	int identity NOT NULL Primary Key
name	Name of this report parameter.	varchar(255) NOT NULL

Every ReportParameter record is associated with the following records:

- ReportRequiredParam
- UserReportParameters

ReportRequiredParam

The ReportRequiredParam table identifies the required parameters in a report.

Column Name	Description	Storage
reportFK	A foreign key to the ID in the Report table.	int NOT NULL

Column Name	Description	Storage
paramFK	A foreign key to the ID in the ReportParameter table.	int NOT NULL

Every ReportRequiredParameter record is associated with the following records:

- Report
- ReportParameter

ReportRole

The ReportRole table identifies the roles associated with a report.

Column Name	Description	Storage
id	Numeric identifier for this file type.	int identity NOT NULL Primary Key
name	Name of the report role.	varchar(255) NOT NULL

Every ReportRole record is associated with the following record:

- ReportRoleScope

ReportRoleScope

The ReportRoleScope table identifies the roles that can view a report.

Column Name	Description	Storage
reportFK	A foreign key to the ID in the Report table.	int NOT NULL
roleFK	A foreign key to the ID in the ReportRole table.	int NOT NULL

Every ReportRoleScope record is associated with the following records:

- Report
- ReportRole

ReportType

The ReportType table identifies the available for a report types.

Column Name	Description	Storage
id	Numeric identifier for this report type.	int identity NOT NULL Primary Key
name	Name of this report type.	varchar(255) NOT NULL

Every ReportType record is associated with the following record:

- Report

ReportUserConfig

The ReportUserConfig identifies who has access to a report.

Column Name	Description	Storage
reportFK	A foreign key to the ID in the Report table.	int NOT NULL
personFK	A foreign key to the ID in the Person table.	int NOT NULL
reportColumnFK	A foreign key to the ID in the ReportColumn table.	int NOT NULL
ordinal	The order number for this ReportUserConfig.	tinyint NOT NULL
userReportFK	A foreign key to the ID in the UserReport table.	int NOT NULL

Every ReportUserConfig record is associated with the following records:

- Person
- Report
- ReportColumn
- UserReport

RequiredFilterParam

A record is inserted in the RequiredFilterParam table that identifies the required parameters for an ACD platform.

Column Name	Description	Storage
id	Numeric identifier for this required filter parameter. This number is auto-generated by the database.	int identity NOT NULL Primary Key
platform	The identity of the ACD platform. The possible values are: • 2–Unified CCX	int NOT NULL
filterParamFK	A foreign key to the ID in the FilterParameter table.	int NOT NULL

Every RequiredFilterParam record is associated with the following record:

- FilterParameter

RetentionData

The RetentionData table is an association between RetentionType and Workflow tables and contains foreign keys to both tables. It contains values for each configured retention period (for example, Archive, HR, Scored, Tagged, Training, or Unscored) for each workflow and each retention type. The RetentionData table contains the values of the retention periods configured for each workflow and each retention type.

Column Name	Description	Storage
workflowFK	A foreign key to the ID in the Workflow table.	int NOT NULL
typeFK	A foreign key to the ID in the RetentionType table.	int NOT NULL
value	The value of the retention period. For Scored and Unscored retention periods, the units are in days. For all other retention periods, the units are in months.	int NOT NULL

Every RetentionData record is associated with the following records:

- Workflow
- RetentionType

RetentionType

A RetentionType contains the available retention types (for example, Archive, HR, Scored, Tagged, Training, or Unscored). Quality Management fills this table after the schema is created.

ID	Name
1	Archive
2	Unscored
3	Scored
4	Pip
5	Tagged
6	Training

The RetentionType table contains the fields shown in the following table

Column Name	Description	Storage
id	Numeric identifier for this retention type.	int identity NOT NULL Primary Key
name	Name of this retention type	nvarchar(128) NOT NULL

Every RetentionType record is associated with the following record:

- RetentionData

Role

The Role table contains a record for each available role in Quality Management.

Column Name	Description	Storage
id	Numeric identifier for this role.	smallint identity NOT NULL Primary Key
name	Name of this role.	varchar(32) NOT NULL

Column Name	Description	Storage
isDashboardAdmin	0—the user is a Dashboard administrator. 1—the user is not a Dashboard administrator.	bit NOT NULL

Every Role record is associated with the following records:

- DashRoleView
- DashWidgetRole
- EvalAlerts
- UI_FeaturePermissions

RtpFilter

The Server table contains the port and IP address information that should be filtered from recording. Use the exclude CAD recording/monitoring traffic.

Column Name	Description	Storage
id	Numeric identifier for this RTP filter.	int identity NOT NULL Primary Key
rtpFilterTypeFK	A foreign key to the ID in the RtpFilterType table.	smallint NOT NULL
value	The IP/Hostname or Port for this filter. **Example:** If the value is 1.1.1.1 all RTP traffic where the value for source or destination is 1.1.1.1 will not be recorded.	nvarchar(256) NOT NULL

Every RtpFilter record is associated with the following record:

- RtpFilterType

RtpFilterType

The RtpFilterType contains records for the possible RTP filter types.

Column Name	Description	Storage
id	Numeric identifier for this RTP filter.	small identity NOT NULL Primary Key
name	Either IP/Hostname or Port.	varchar[32] NOT NULL

Every RtpFilterType record is associated with the following record:

- RtpFilter

ScreenMonitoring

Quality Management creates a new record in the ScreenMonitoring table every time a agent logs into their desktop where Desktop Record service is running.

Column Name	Description	Storage
personFK	A foreign key to the ID in the Person table.	int NOT NULL
invite	The invite sent from the Desktop Recording service on the agent's client machine.	varchar(1500)
password	The password associated with the invite.	varchar(1500)
LastModified	The last time this record was modified.	datetime (16,3) NOT NULL

Every Screen record is associated with the following records:

- Person

Server

The Server table contains records for each configured server.

Column Name	Description	Storage
id	Numeric identifier for this server. This number is auto-generated by the database.	bigint identity NOT NULL Primary Key

Column Name	Description	Storage
serverTypeFK	A foreign key to the ID in the ServerType table.	bigint NOT NULL
ipHostName	The name of the host.	varchar(40) NOT NULL

Every Server record is associated with the following records:

- ACDServer
- EventLogging
- GamificationConnectionInfo
- Gateway
- MediaFile
- RecordingClusterServer
- ServerProperties
- ServerType
- SignalingGroup
- SignalingGroupServer
- SiteServer
- TelephonyGroupServer

ServerProperties

The ServerProperties table specifies the properties for each configured server.

Column Name	Description	Storage
serverFK	A foreign key to the ID in the Server table.	bigint NOT NULL
id	Numeric identifier for ServerProperties. This number is auto-generated by the database.	varchar(32) NOT NULL

Column Name	Description	Storage
setting	The id property setting. **Example:**If id is port, the setting might be 59010.	nvarchar (1073741823) NOT NULL

Every ServerProperties record is associated with the following records:

- Server

ServerType

The ServerType table specifies the type of server for each configured server.

Column Name	Description	Storage
id	Numeric identifier for this server. This number is auto-generated by the database.	bigint identity NOT NULL Primary Key
name	The name of the server type.	varchar[32] NOT NULL

Every ServerType record is associated with the following record:

- Server

SignalingGroup

The SignalingGroup table specifies the type of server for each configured server.

Column Name	Description	Storage
id	Numeric identifier for this signaling group. This number is auto-generated by the database.	int identity NOT NULL Primary Key
name	The name of the server type.	varchar[64] NOT NULL
telephonyGroupFK	A foreign key to the ID in the telephonyGroup table.	int NOT NULL

Column Name	Description	Storage
primarySignalingFK	A foreign key to the ID in the Server table.	bigint NOT NULL
backupSignalingFK	The priority for the signaling group.	bigint NOT NULL

Every SignalingGroup record is associated with the following record:

- RecordingCluster
- Server
- SignalingGroupServer
- TelephonyGroup

SignalingGroupServer

The SingalingGroupServer table specifies the type of server for each configured server.

Column Name	Description	Storage
signalingGroupFK	A foreign key to the ID in the signalingGroup table.	int NOT NULL
serverFK	A foreign key to the ID in the server table.	bigint NOT NULL
signalingAssociation		smallint NOT NULL
priority		smallint NOT NULL

Every SingalingGroupServer record is associated with the following record:

- Server
- SignalingGroup

Site

The Site table contains records for each configured site.

Column Name	Description	Storage
id	The unique ID of the site.	int identity NOT NULL Primary Key
name	The user-defined name for the site.	varchar (2147483647) NOT NULL
enableUpdates	Indicates whether to turn on True Update for clients or not. By default this is set to False.	bit NOT NULL
isDefault	Indicates if this is the default site. By default this is set to False.	bit NOT NULL
isTwoStageUploadEnableed	Indicates if Two Stage Upload is enabled. By default this is set to False.	bit NOT NULL

Every Site record is associated with the following records:

- Ccr
- RecordingCluster
- SiteServer
- Team

SiteServer

The SiteServer table specifies which servers belong to a site.

Column Name	Description	Storage
siteFK	The foreign key to the Site table.	int NOT NULL
serverFK	The foreign key to the Server table.	bigint NOT NULL

Every SiteServer record is associated with the following records:

- Site
- Server

Supervisor

Quality Management creates a Supervisor record every time a team is assigned to be supervised by a person with a supervisor role in Quality Management Administrator and/or for every supervisor of a team at the time a supervisor is synced from the ACD. This means that when a supervisor supervises two teams, two records are created.

After being synced or saved in Quality Management Administrator, it will be at most ten minutes until the Monitoring and Recording Sync service adds this record to the database. Activation and deactivation dates reflect the time of the database sync that created or deactivated the record.

Column Name	Description	Storage
agentFK	A foreign key to the ID of the supervisor in the Agent table.	int NOT NULL
teamFK	A foreign key to the ID of the team being supervised in the Team table.	int NOT NULL
activated	The GMT date and time when the supervisor role was assigned to the person or the ACD supervisor was configured.	datetime (16,3) NOT NULL
deactivated	The GMT date and time when the supervisor role was removed from the person or the person was deleted or unconfigured.	datetime (16,3) NOT NULL
lastModified	The last time the supervisor was modified.	datetime (16,3) NOT NULL

Every Supervisor record is associated with the following records:

- Agent
- Team

Survey

The Survey table specifies the type of server for each configured server.

Column Name	Description	Storage
id	Numeric identifier for this survey. This number is auto-generated by the database.	bigint int identity NOT NULL Primary Key

Column Name	Description	Storage
callIdentifier	The ID to associate related contacts. This is used to display associated calls in the Quality Management in Unified Workforce Optimization, so various legs of the same customer contact can be viewed together.	varchar[52] NOT NULL
surveyFormFK	A foreign key to the ID in the surveyForm table.	int NOT NULL
score	The score for the survey.	int NOT NULL
lastModified	The last time the survey was modified.	datetime (16,3) NOT NULL

Every Survey record is associated with the following record:

- SurveyForm
- SurveyResults

SurveyForm

Quality Management adds a record to the SurveyForm table each time a new survey form is created.

Column Name	Description	Storage
id	Numeric identifier for this evaluation form. This identifier is auto-generated by the database.	int identity NOT NULL Primary Key
formId	The ID associated with the form.	int NOT NULL
name	Name of this survey form. Names must be unique.	nvarchar(128) NOT NULL
description	Description of this survey form.	nvarchar(256) NOT NULL
creator	Creator of the survey form.	nvarchar(128) NOT NULL

Column Name	Description	Storage
created	Date and time this survey form was created, in GMT.	datetime (16,3) NOT NULL
totalPoints	The total points assigned to this survey.	int NOT NULL
statusFK	A foreign key into the status table.	tinyint NOT NULL

Every SurveyForm record is associated with the following records:

- Survey
- SurveyFormStatus
- SurveyQuestion

SurveyFormStatus

Quality Management adds a record to the SurveyFormStatus table each time the status is updated.

Column Name	Description	Storage
id	Numeric identifier for this evaluation form. This identifier is auto-generated by the database.	int NOT NULL Primary Key
status	The status of the survey form.	nvarchar[48] NOT NULL

Every SurveyFormStatus record is associated with the following record:

- SurveyForm

SurveyQuestion

Quality Management adds a record to the SurveyQuestion table, each time a new question is created in a survey form.

20	Description	Storage
id	Numeric identifier for this survey question, auto-generated by the database.	int identity NOT NULL Primary Key

20	Description	Storage
surveyFormFK	A foreign key to the ID in the SurveyForm table.	int NOT NULL
weight	The weight of this section. Max = 1.0.	float NOT NULL
questionNumber	The number associated with the question.	tinyint NOT NULL
question	The question associated with the survey.	nvarchar(1024) NOT NULL
type	The type of question.	nvarchar(64) NOT NULL

Every SurveyQuestion record is associated with the following record:

- SurveyForm

SurveyResults

Quality Management adds a record to the SurveyResults table, for each survey result.

Column Name	Description	Storage
id	Numeric identifier for this survey result, auto-generated by the database.	bigint identity NOT NULL Primary Key
surveyFK	A foreign key to the ID in the Survey table.	bigint NOT NULL
questionNumber	The number associated with the question.	tinyint NOT NULL
result	The answer to the question.	nvarchar(1024) NOT NULL
receivedWeight	The weigh associated with the question.	float NOT NULL

Every SurveyQuestion record is associated with the following records:

- Survey

SyncFilter

The SyncFilter table contains information about the synchronization filter.

Column Name	Description	Storage
id	Numeric identifier for the ACD, auto-generated by the database.	int identity NOT NULL Primary Key

Every SyncFilter record is associated with the following record:

- FilterProperties
- SyncFilter
- SyncFilterTelephony

SyncFilterACD

The SyncFilterACD table contains information about the ACD sync filter.

Column Name	Description	Storage
syncFilterFK	A foreign key to the ID in the SyncFilter table.	int NOT NULL
acdFK	A foreign key to the ID in the ACD table.	int NOT NULL

Every SyncFilterACD record is associated with the following records:

- syncFilter
- ACD

SyncFilterTelephony

The SyncFilterTelephony table contains information about the telephony sync filter.

Column Name	Description	Storage
syncFilterFK	A foreign key to the ID in the SyncFilter table.	int NOT NULL

Column Name	Description	Storage
telephonyGroupFK	A foreign key to the ID in the TelephonyGroup table.	int NOT NULL

Every SyncFilterACD record is associated with the following records:

- syncFilter
- TelephonyGroup

Team

The Team table contains information about specific teams. A record is created when a knowledge worker team is created in the Quality Management Administrator or when a team is synced from the ACD. After being synced or saved in Quality Management Administrator, it will be at most ten minutes until the Monitoring and Recording Sync service adds this record to the database. Activation and deactivation dates reflect the time of the database sync that created or deactivated the record.

Column Name	Description	Storage
id	Numeric identifier for the team. This number is auto-generated by the database.	int identity NOT NULL Primary Key
name	Name of the team.	nvarchar(120) NOT NULL
displayId	The team identifier displayed in Monitoring and Recording Administrator. For more information, see Dates in the Database.	nvarchar(64) NOT NULL
realmFK	A foreign key to the Realm table for whether this is a team created in Quality Management or synced from the ACD.	tinyint NOT NULL
activated	The GMT date and time when the team was created (or synced from the ACD).	datetime (16,3) NOT NULL
deactivated	The GMT date and time when the team was deleted.	datetime (16,3) NOT NULL
archiveWorkflowFK	A foreign key to the ID in the Workflow table.	int NOT NULL

Column Name	Description	Storage
qualityWorkflowFK	A foreign key to the ID in the Workflow table.	int NOT NULL
siteFK	A foreign key to the ID in the Site table.	int NOT NULL
lastModified	The last time the team was modified.	datetime (16,3) NOT NULL
acdFK	A foreign key to the ID in the ACD table.	int NOT NULL

Every Team record is associated with the following records:

- ACD
- AgentTeam
- Ccr
- Evaluator
- Site
- Supervisor
- TeamGroup
- WorkflowRuleWhoTeam
- Workflow

TeamGroup

Quality Management creates a TeamGroup record every time a team is assigned to a group in Quality Management Administrator.

After being synced or saved in Quality Management Administrator, it will be at most ten minutes until the Monitoring and Recording Sync service adds this record to the database. Activation and deactivation dates reflect the time of the database sync that created or deactivated the record.

Column Name	Description	Storage
teamFK	A foreign key to the ID in the Team table.	int NOT NULL

Column Name	Description	Storage
groupFK	A foreign key to the ID in the Groups table.	int NOT NULL
activated	The GMT date and time when the team was assigned to the group.	datetime (16,3) NOT NULL
deactivated	The GMT date and time when the team was removed from the group.	datetime (16,3) NOT NULL

Every TeamGroup record is associated with the following records:

- Team
- Groups

TelephonyGroup

The TelephonyGroup table is a telephony entity that provides a unique set of phones and telephony devices. It also includes a number of telephony servers, including at least one Quality Management CTI server.

Column Name	Description	Storage
id	Numeric identifier for the telephony group. This number is auto-generated by the database.	int identity NOT NULL Primary Key
inclusionList	The inclusion list XML for this telephony group (cluster).	nvarchar (1073741823) NOT NULL
typeFK	A foreign key to the ID in the TelephonyGroupType table.	int NOT NULL
signalingMethod	The signaling method in user. Either CTI signaling, MediaSense signaling, or both.	smallint NOT NULL
name	Name of this telephony group.	varchar(64) NOT NULL

Every TelephonyGroup record is associated with the following records:

- Ccr
- SignalingGroup
- SyncFilterTelephony
- TelephonyGroupType
- TelephonyGroupServer
- VoiPMonitorDevice

TelephonyGroupServer

The TelephonyGroupServer table associates servers with telephony groups.

Column Name	Description	Storage
telephonyGroupFK	A foreign key to the ID in the TelephonyGroup table.	int NOT NULL
serverFK	A foreign key to the ID in the Server table.	bigint NOT NULL

Every TelephonyGroupServer record is associated with the following records:

- Server
- TelephonyGroup

TelephonyGroupType

The TelephonyGroupType table indicates the telephony group type associated with a TelephonyGroup.

Column Name	Description	Storage
id	Numeric identifier for this telephony group type. This number is auto-generated by the database.	int NOT NULL
name	The name of the telephony group type.	varchar(25) NOT NULL
recordingSide	The recording method fro this telephony group type. The options are as follows: • Gateway Recording • Agent Recording	varchar(25) NOT NULL

Every TelephonyGroupType record is associated with the following record:

- TelephonyGroup

Timezone

The Timezone table is filled with default time zones by the Quality Management. If other time zones are used, they are added to this table.

Column Name	Description	Storage
id	The numeric identifier for this time zone. This number is auto-generated by the database.	tinyint identity NOT NULL Primary Key
javaName	Name of the Java time zone.	varchar(255) NOT NULL

Every Timezone record is associated with the following records:

- ArchiveAudit
- Ccr
- Eval
- EvalComment
- EvalForm
- EventAudit

UI_Feature

Quality Management loads the UI_Feature table with the valid features.

Column Name	Description	Storage
id	Numeric identifier for this feature. This number is auto-generated by the database.	int identity NOT NULL Primary Key
featureGroupFK	A foreign key to the ID in the UI_FeatureGroup table.	smallint NOT NULL

Column Name	Description	Storage
name	The name of this feature. **Example:** dashboard or searchAndPlay	nvarchar(128) NOT NULL
useHttps	0–pass the URL using HTTP or HTTPS protocol and port. 1–do not pass the URL using HTTP or HTTPS protocol and port.	bit NOT NULL
uri	URL for the feature's Google Gadget XML	nvarchar(1024) NOT NULL
icon	The icon associated with this feature.	nvarchar(64) NOT NULL
ordinal	The order number for this feature.	int NOT NULL
requiresLicensedUser	0–user requires a license to use the features. 1–user does not require a license to use the features.	bit NOT NULL

Every UI_Feature record is associated with the following records:

- Field
- Filter
- UI_FeatureGroup
- UI_FeaturePermissions

UI_FeatureGroup

The UI_FeatureGroup table provides a unique set of features groups.

Column Name	Description	Storage
id	Numeric identifier for this feature group. This number is auto-generated by the database.	smallint identity NOT NULL Primary Key

Column Name	Description	Storage
name	Name of this feature group.	nvarchar(64) NOT NULL

Every UI_FeatureGroup record is associated with the following records:

- UI_Feature

UI_FeaturePermissions

The UI_FeaturePermissions contains the feature permissions for each UI_Feature based on Role and License.

Column Name	Description	Storage
featureFK	A foreign key to the id in the UI_Feature table.	int NOT NULL
roleFK	A foreign key to the id in the Role table.	smallint NOT NULL
permission	0–if the role can access the feature. 1–if the role cannot access the feature.	bit NOT NULL

Every UI_FeaturePermissions record is associated with the following records:

- UI_Feature
- Role

UniqueAdminGroup

The UniqueAdminGroup table generates unique adminGroupNum values for the Group table and generates unique IDs for newly created knowledge workers. Both Quality Management Administrator and Monitoring and Recording Sync service use this table to keep groups and knowledge workers synced to the Quality Management database.

Column Name	Description	Storage
id	The unique ID that has been used as the last adminGroupNum in the Groups table. This number is auto-generated by the database.	int NOT NULL Primary Key

UploadState

Quality Management loads the UploadState table with default upload state values after the database schema is created. Each record in this table is a possible state for an audio or screen file Ccr record. The default values are shown in the following table.

ID	Name
0	No file to upload
1	File not yet uploaded
2	File is uploaded

The UploadState table contains the fields shown in the following table.

Column Name	Description	Storage
id	Numeric identifier of the upload state.	tinyint NOT NULL Primary Key
name	Name of the upload state.	varchar(32) NOT NULL

Every UploadState record is associated with the following records:

- Ccr
- MediaFile

UserReport

The UserReport contains information on each user report.

Column Name	Description	Storage
id	Numeric identifier for this user report.	int identity NOT NULL Primary Key
reportFK	A foreign key to the ID in the Report table.	int NOT NULL
personFK	A foreign key to the ID in the Person table.	int NOT NULL

Column Name	Description	Storage
reportName	Name of this report.	varchar(255) NOT NULL

Every UserReport record is associated with the following records:

- Person
- Report
- ReportUserConfig
- UserReportParameters

UserReportParameters

The UserReportParameters identifies the parameters for the user report.

Column Name	Description	Storage
id	Numeric identifier for theses user report parameters.	int identity NOT NULL Primary Key
userReportFK	A foreign key to the ID in the UserReport table.	int NOT NULL
reportParameterFK	A foreign key to the ID in the ReportParameter table.	int NOT NULL
settingValue	The value of the parameter.	varchar (1073741823) NOT NULL

Every UserReportParameters record is associated with the following records:

- UserReport
- ReportParameter

VoiPMonitorDevice

The VoiPMonitorDevice table contains configuration information for each VoIP device.

Column Name	Description	Storage
id	Numeric identifier for the VoIP monitoring device. This number is auto-generated by the database.	bigint identity NOT NULL Primary Key
personFK	A foreign key to the id in the Person table.	int NOT NULL
deviceName	The name of the device.	varchar[64] NOT NULL
deviceTypeFK	The device type.	int NOT NULL
monitorServerFK	A foreign key to the id in the Server table.	bigint NOT NULL
recordingTypeFK	A foreign key to the id in the VoiPMonitorDeviceType table.	int NOT NULL
telephonyGroupFK	A foreign key to the id in the TelephonyGroup table.	int NOT NULL
virtualExtFK	A foreign key to the id in the VoiPMonitorDevice table.	bigint NOT NULL
loggedInPersonFK	A foreign key to he loggedInPerson table.	int NOT NULL
recordingClusterFK	A foreign key to the RecordingCluster table.	int NOT NULL
recordingTones	0–the beep tones are not enabled for the device. 1–the beep tones are enabled for the device.	bit NOT NULL

Every VoiPMonitorDevice record is associated with the following records:

- Person
- RecordingCluster
- RecordingType
- TelephonyGroup

- VoiPMonitorDeviceLine
- VoiPMonitorDeviceType

VoiPMonitorDeviceLine

The VoiPMonitorDeviceLine table maps the devices to extensions and partitions. A device might have multiple extensions and partitions.

Column Name	Description	Storage
id	Numeric identifier for the VoIP monitoring device line. This number is auto-generated by the database.	bigint identity NOT NULL Primary Key
deviceFK	A foreign key to the id in the VoiPMonitorDevice table.	bigint NOT NULL
extension	The extension for this device.	varchar(32) NOT NULL
partition	The partition for this device.	varchar(50) NOT NULL

Every VoiPMonitorDeviceLine record is associated with the following records:

- VoiPMonitorDevice

VoiPMonitorDeviceType

The VoiPMonitorDeviceType table indicates the device type associated with a VoiPMonitorDevice.

Column Name	Description	Storage
id	Numeric identifier for the VoIP monitoring device type. This number is auto-generated by the database.	int NOT NULL Primary Key
name	The name of this VoIP monitoring device type **Example:** Phone	varchar(25) NOT NULL

Every VoiPMonitorDeviceType record is associated with the following record:

- VoiPMonitorDevice

vw_ActiveAgentsInGroups

Quality Management creates a new record for active agents in groups.

Column Name	Description	Storage
group_id	Numeric identifier for this group.	int NOT NULL
team_id	Numeric identifier for this team.	int NOT NULL
tm_agent_id	Numeric identifier for this the team agent.	int NOT NULL
tm_person_id	Numeric identifier for this the team person.	int NOT NULL

vw_ActiveAgentsInTeams

Quality Management creates a new record for active agents in teams.

Column Name	Description	Storage
team_id	Numeric identifier for this team.	int NOT NULL
tm_agent_id	Numeric identifier for this the team agent.	int NOT NULL
tm_person_id	Numeric identifier for this the team person.	int NOT NULL

vw_ContactsWithActiveOrg

A vw_ContactsWithActiveOrg record is used for contact reconciliation.

Column Name	Description	Storage
skillTargetId	Realm.skillTargetID is the unique identifier used for a person. The skillTargetId is the unique ID retrieved from the ACD database for agents or auto-generated for knowledge workers in Quality Management.	nvarchar(120) NOT NULL
firstname	The first name of the this person.	nvarchar(120) NOT NULL

Column Name	Description	Storage
lastname	The last name of the this person.	nvarchar(120) NOT NULL
teamId	Numeric identifier for this team.	int NOT NULL
teamName	The name of the team.	nvarchar(120) NOT NULL
groupId	Numeric identifier for this group.	int NOT NULL
groupName	The name of the group.	nvarchar(64) NOT NULL
startTime	The local start time of the call in the timezone where the call occurred. The localStartTime is used by Quality Management Reporting to compare the start time with the local date range selections when viewing reports.	datetime (16,3) NOT NULL
pathName	The name of the path.	nvarchar(510) NOT NULL
serverName	The name of the server.	varchar(40) NOT NULL
Offset	The time the recording starts relative to the startTime timestamp of the Ccr, in milliseconds. This is also the time between the CTI ringing event and the CTI established event.	int NOT NULL
Duration	The duration of the recording in milliseconds.	int NOT NULL
ccrId	A unique ID used in the Ccr table.	bigint NOT NULL
ani	The originating phone number for this contact.	nvarchar(128) NOT NULL

Column Name	Description	Storage
dnis	The called phone number for this contact.	nvarchar(128) NOT NULL
icmCallId	The Peripheral Call Key (or Call ID) for the call from the CCM system.	varchar(128) NOT NULL
startTimeTzFK	A foreign key into the Timezone table. A value of zero indicates that the time zone is currently unknown. An unknown time zone might be updated at some point. For more information, see Dates in the Database.	tinyint NOT NULL
qualityReasonFK	A foreign key to the ID in the RecordingReason table. The reason why the cal was or was not marked for quality management purpose.	smallint NOT NULL
localStartTime	The local start time of the call in the timezone where the call occurred. The localStartTime is used by Quality Management Reporting to compare the start time with the local date range selections when viewing reports.	datetime (16, 3) NOT NULL
personId	The ID of the agent.	int NOT NULL
extension	The extension(s) where the login or logout occurred.	varchar(16) NOT NULL
line	Agent's extension.	nvarchar(128) NOT NULL
ccrType	The type of contact. The contact type is either call or non-call	varchar [21474�83647] NOT NULL
recordingType	The VoIP monitor recording type.	varchar[25] NOT NULL
qualityReason	The reason why the call was or was not marked for quality management purposes.	varchar[64] NOT NULL
archiveReason	The reason why the cal was or was not marked for archiving.	varchar[64] NOT NULL

vw_PersonDetails

Quality Management creates a new record for person details. It is used with Ccr to ensure the person has active states for agent, agent team, team, and group.

Column Name	Description	Storage
person_id	Numeric identifier for this person.	int NOT NULL
Person_ lastName	The last name of the this person.	nvarchar(120) NOT NULL
Person_ firstName	The first name of the this person.	nvarchar(120) NOT NULL
Person_username	The username for this person.	nvarchar(128) NOT NULL
Person_ domainName	The domain name for this person.	nvarchar(128) NOT NULL
Person_displayId	The display ID for this person.	nvarchar(120) NOT NULL
Team_id	Numeric identifier for this team.	int NOT NULL
Team_name	The name of the team	nvarchar(120) NOT NULL
Team_displayId	The display ID for this team.	nvarchar(64) NOT NULL
Groups_id	The ID for this group.	int NOT NULL
Groups_name	The name of the group.	nvarchar(64) NOT NULL

vw_WfmAgentReportCard

Quality Management creates a new record for the WFM agent report card.

Column Name	Description	Storage
ICMAGENT	The identity of the agent.	nvarchar(120) NOT NULL
STARTTIME	The time and date when the report card began.	datetime (16,3) NOT NULL
TOTALSCORE	The score for this evaluation. Null if the evaluation has been claimed but no questions have yet been scored.	float NOT NULL
FORMID	Numeric identifier for this evaluation form. This identifier is auto-generated by the database.	int NOT NULL

Workflow

The Workflow table specifies the workflows that are configured in Quality Management. The workflow contains a reference to a workflowXML that is generated by the Quality Management Administrator.

Column Name	Description	Storage
id	The primary key for the workflow.	int identity NOT NULL Primary key
typeFK	A foreign key reference to ID in the WorkflowType table.	smallint NOT NULL
isActive	True—the workflow is active and available. False—the workflow was deleted and the record still exists.	bit NOT NULL
name	Identifies the name of the workflow.	nvarchar[128] NOT NULL
canChangeForm	0—if the evaluator can change the form assigned to a contact. 1—if the evaluator cannot change the evaluation form assigned to a contact.	bit NOT NULL
endOfDay	The time when the recording is uploaded.	nvarchar[10] NOT NULL

Column Name	Description	Storage
immediateUpload	0–recording is immediately uploaded after the recording is complete. 1–the recording is uploaded based on normal upload setting rules.	bit NOT NULL
extendScreen	The number of seconds screen recording continues to record after-call work after a call is dropped.	int NOT NULL
lastProcessDurationSec	The number of seconds elapsed during processing of workflow.	int NOT NULL
immediateScreenUpload	1–the workflow was marked for immediate screen upload. 0–the workflow was not marked for immediate screen upload.	bit NOT NULL

Every Workflow record is associated with the following records:

- Ccr
- RetentionData
- Team
- WorkflowType
- WorkflowClassifier

WorkflowClassifier

The WorkflowClassifier table contains records for each workflow classifier.

Column Name	Description	Storage
id	The primary key for the workflow classifier.	int identity NOT NULL Primary key
typeFK	A foreign key reference to ID in the WorkflowClassifierType table.	int NOT NULL
workflowFK	A foreign key reference to ID in the Workflow table.	int NOT NULL

Column Name	Description	Storage
evalFormFK	A foreign key reference to ID in the EvalForm table.	int NOT NULL
name	Identifies the name of the workflow classifier.	nvarchar[128] NOT NULL
ordinal	The order number for this workflow classifier.	int NOT NULL
logging	0–if logging is enabled. 1–if logging is not enabled.	bit NOT NULL
uploadScreen	0–if the screen recording is uploaded. 1–if the screen recording is not uploaded.	bit NOT NULL

Every WorkflowClassifier record is associated with the following records:

- EvalForm
- Workflow
- WorkflowClassifierNumber
- WorkflowClassifierType
- WorkflowRule

WorkflowClassifierNumber

The WorkflowClassifierNumber table contains records for each workflow classifier number.

Column Name	Description	Storage
id	The primary key for the workflow classifier number.	int identity NOT NULL Primary key
classifierFK	A foreign key reference to ID in the WorkflowClassifier table.	int NOT NULL
number	The number for the workflow classifier.	nvarchar(128) NOT NULL

Column Name	Description	Storage
ordinal	The order number for this workflow classifier number.	int NOT NULL
numberTypeFK	A foreign key reference to ID in the WorkflowClassifierNumberType table.	tinyint NOT NULL

Every WorkflowClassifierNumber record is associated with the following record:

- WorkflowClassifier
- WorkflowClassifierNumberType

WorkflowClassifierNumberType

The WorkflowClassifierNumberType table contains records for each workflow classifier number.

Column Name	Description	Storage
id	The primary key for the workflow classifier number type.	tinyint NOT NULL
name	The name of the workflow classifier.	varchar(10) NOT NULL

Every WorkflowClassifierNumber record is associated with the following record:

- WorkflowClassifier

WorkflowClassifierType

The WorkflowClassifierType table contains records for each workflow classifier type.

Column Name	Description	Storage
id	The primary key for the workflow classifier type.	int identity NOT NULL Primary key
name	Identifies the name of the workflow classifier type.	nvarchar(40) NOT NULL

Every WorkflowClassifierType record is associated with the following record:

- WorkflowClassifier

WorkflowRule

The WorkflowRule table contains records for each workflow classifier rule.

Column Name	Description	Storage
id	The primary key for the workflow classifier.	int identity NOT NULL Primary key
name	Identifies the name of the workflow rule.	nvarchar[128] NOT NULL
classifierFK	A foreign key reference to ID in the WorkflowClassifier table.	int NOT NULL
whoTypeFK	A foreign key reference to ID in the WorkflowRuleWhoType table.	int NOT NULL
whenTypeFK	A foreign key reference to ID in the WorkflowRulesWhenType table.	int NOT NULL
ordinal	The order number for this workflow rule.	int NOT NULL

Every WorkflowRule record is associated with the following records:

- WorkflowClassifier
- WorkflowRuleWhoType
- WorkflowRuleWhenType
- WorkflowRuleWhat
- WorkflowRuleWhenRange
- WorkflowRuleWhenWeekly
- WorkflowRuleWhoPerson
- WorkflowRuleWhoTeam

WorkflowRuleWhat

The WorkflowRuleWhat table contains records for each WHAT rule in a workflow.

Column Name	Description	Storage
id	The primary key for the workflow classifier.	int identity NOT NULL Primary key
ruleFK	A foreign key reference to ID in the WorkflowRule table.	int NOT NULL
reasonFK	A foreign key reference to ID in the RecordingReason table.	smallint NOT NULL
minDuration	The minimum number of seconds allowed per call.	int NOT NULL
callsPerDay	The number of calls per day to record.	int NOT NULL

Every WorkflowRuleWhat record is associated with the following records:

- RecordingReason
- WorkflowRule
- WorkflowRuleWhatPeriod

WorkflowRuleWhatPeriod

The WorkflowRuleWhatPeriod table contains records for each period in a WHAT rule.

Column Name	Description	Storage
whatFK	A foreign key reference to ID in the WorkflowRuleWhat table.	int NOT NULL
startTime	The time and date when the period began.	nvarchar(10) NOT NULL
endTime	The time and date when the period ended.	nvarchar(10) NOT NULL

Every WorkflowRuleWhatPeriod record is associated with the following record:

- WorkflowRuleWhat

WorkflowRuleWhenRange

The WorkflowRuleWhenRange table contains records for each range in a WHEN rule.

Column Name	Description	Storage
id	The primary key for the workflow when range.	int identity NOT NULL Primary key
ruleFK	A foreign key reference to ID in the WorkflowRule table.	int NOT NULL
beginDate	The date and time when the range began.	datetime (16,3) NOT NULL
endDate	The date and time when the range ended.	datetime (16,3) NOT NULL

Every WorkflowRuleWhenRange record is associated with the following record:

- WorkflowRule

WorkflowRuleWhenType

The WorkflowRuleWhatType table contains records for each type in a WHEN rule.

Column Name	Description	Storage
id	The primary key for the workflow rule WHEN type.	int identity NOT NULL Primary key
name	The name of the workflow rule WHEN type.	nvarchar(20) NOT NULL

Every WorkflowRuleWhatType record is associated with the following record:

- WorkflowRule

WorkflowRuleWhenWeekly

The WorkflowRuleWhenWeekly table contains records for each WHEN rule with a Select When of Weekly.

Column Name	Description	Storage
RuleFK	A foreign key reference to ID in the WorkflowRule table.	int NOT NULL
weeklyTypeFK	A foreign key reference to ID in the WorkflowRuleWeeklyType table.	int NOT NULL

Every WorkflowRuleWhenWeekly record is associated with the following records:

- WorkflowRule
- WorkflowRuleWhenWeeklyType

WorkflowRuleWhenWeeklyType

The WorkflowRuleWhenWeeklyType table contains records for the day of the week

Column Name	Description	Storage
id	The primary key for the workflow rule WHEN weekly type.	int NOT NULL Primary key
name	The day of the week (1-7, where 1 = Sunday and 7 = Saturday).	nvarchar(20) NOT NULL

Every WorkflowRuleWhenWeeklyType record is associated with the following record:

- WorkflowRuleWhenWeekly

WorkflowRuleWhoPerson

The WorkflowRuleWhoPerson table contains records for each a person associated with a WHO rule.

Column Name	Description	Storage
ruleFK	A foreign key reference to ID in the WorkflowRule table.	int NOT NULL
personFK	A foreign key reference to ID in the Person table.	int NOT NULL

Every WorkflowRuleWhoPerson record is associated with the following records:

- WorkflowRule
- Person

WorkflowRuleWhoTeam

The WorkflowRuleWhoTeam table contains records for each team associated with a WHO rule.

Column Name	Description	Storage
ruleFK	A foreign key reference to ID in the WorkflowRule table.	int NOT NULL
teamFK	A foreign key reference to ID in the Team table.	int NOT NULL

Every WorkflowRuleWhoTeam record is associated with the following records:

- WorkflowRule
- Team

WorkflowRuleWhoType

The WorkflowRuleWhoType table contains records for each type for a WHO rule in a workflow.

Column Name	Description	Storage
id	The primary key for the workflow rule WHO type.	int NOT NULL Primary key
name	The name of the workflow rule WHO type.	nvarchar(20) NOT NULL

Every WorkflowRuleWhoType record is associated with the following record:

- WorkflowRule

WorkflowType

The WorkflowType table specifies the available workflow types. Quality Management fills this table after the schema is created. The default values are shown in the following table.

ID	Name
1	Archive
2	Quality

The WorkflowType table contains the fields shown in the following table.

Column Name	Description	Storage
id	The primary key for the workflow.	smallint identity NOT NULL Primary key
name	The name of the workflow type.	varchar(32) NOT NULL

Every WorkflowType record is associated with the following record:

- Workflow

Database Change Log

The following tables provide a historical log of changes to the database schema for Quality Management. If you have an application that relies on data in the database schema, use this log to verify if the latest changes affect your application.

Database Change Log for Schema 6

Schema Version	Description
6.907	Remove screenMaxDuration and change voiceMaxDuration to recordingMaxDuration
6.904	Add index to LoginState table
6.903	Update retention data types for existing archive workflows to use new explicit "Tagged (Archive)" data type
6.902	Add mapping of Eval Goal actions and Roles
6.901	Added gamification ordinal column
6.898	Added covering index on RetentionData (workflowFK, typeFK, unitTypeFK) to improve DB Cleaner retention query

Schema Version	Description
6.897	Added "Tagged (Archive)" retention type to RetentionType table for Customizable Retention Policies feature
6.896	Added schema changes for retention rules implementation (part of the Customizable Retention Policies feature). Tables: RetentionRule, RetentionRuleCondition, RetentionRuleData
6.891	Changes for Interaction Collaboration feature
6.869	Update UserReport and related foreign keys to CASCADE on delete
6.865	Update QMAdminAudit to make jsonDiff and stringDiff columns NVARCHAR(max)
6.863	Added unitTypeFK to RetentionData, and added RetentionUnitTypes table, and did months to days conversion on existing RetentionData values
6.858	Add lastModified column to Media and MediaFile tables
6.854	Adding isTranscribed field to CCR
6.845	Add Survey Indexes
6.844	Add AcdTelephonyGroup relationship table
8.842	Increase size of Server.ipHostName column from 40 to 256.
6.841	Add isAnalytics column to MetaDataField and set its default value to 0
6.840	Realm removal
6.836	New Survey Feature Support
6.792	Add recordScreen column to Workflow table and set the initial upgrade value based on the workflow type.
6.788	Changed Gamification Metric label from adherence to Adherence.
6.785	Removed the level parameter from reports as it is not used any more.
6.784	Added activeCallOnly column to the RecordingApiCommand table.
6.783	Added new Event Logging and related tables

Schema Version	Description
6.780	Cleaned up unused EventAudit and related tables
6.776	Upgrade existing ACDFilter data into new SyncFilter data structures.
6.769	Update Person and Team constraints to use acdFK instead of realmFK to support multiple ACDs of a single type.
6.765	Change ACDFilter to SyncFilter and split the functionality into ACD and Telephony filter types.
6.759	Added ACD references to tables that have data synced over from an ACD.
6.750	Created new 'UCCX Server' type and upgraded any existing servers to use that type.
6.748	Created ACD data structure to allow multiple groupings of ACD servers.
6.737	Changes to upgrade changed package names for some CTI implementations and SIP implementations.
6.735	Changes to support Evaluator Queue Performance Metrics (Eval Goals).
6.717	Two Stage Upload support.
6.716	AudioCodes port change.
6.674	Changes to support Self Evaluation.
6.663	Forcing the VoIP Monitor Device recording type to None where device type is User Profile.
6.662	Set encodeFramesPerSecond to 1 instead of 4 for 9.4 SR1 upgrades.
6.661	Set encodeMFBitrate to 200 on 9.4 SR1 upgrades.
6.659	Added ReconciliationHistory Status table with description of the person match status code.
6.658	Updated AudioCodes Retrieval Service port.
6.656	Clean up old, unused columns in CDR table.
6.649	Set all .rec and .erec recordings to .em4v recording type.

Schema Version	Description
6.648	Added license type to Ccr.
6.643	Removing Administrator dashboard.
6.641	Setting HDS server suffix.
6.640	Adding isAxlProvider.
6.593	WebM Schema Changes.
6.583	Evaluator Type Work.
6.581	Adding column to UI_Feature table to indicate if a license is needed for the application.
6.580	Rename associatedCallId column to callIdentifier. (This column can now contain associatedCallId, Contact ID, ICM Call ID, or metadata).
6.569	Making Admin Roles true roles.
6.567	Eval Comment changes for section and question comments.
6.564	Dropped the insert trigger on ArchiveAudit table.
6.563	Added the eventId bigint column to GamificationEvent.
6.562	Added the isDataPushed bit column to GamificationEventType.
6.561	Added Application Management to the list of UI features.
6.560	Added a unique constraint to GamificationLevelScoring for combined columns gamificationLevelFK and minScoreThreshold.
6.559	Added a unique constraint to GamificationPointRangeScoring for combined columns gamificationPointRangeFK and rangeMinimum.
6.555	Added support for Admin Auditing.
6.541	6.541 Added support for Gamification.
6.537	Added support for the Survey Form.
6.531	Added support for Admin Partitioning.
6.513	Created Signaling Groups.
6.510	Added the MetaDataMappableColumms table, added mappedColumnFK column to the MetaData table

Schema Version	Description
6.509	Created Foreign Key constraint for telephonyGroupFK.
6.508	Created a new column in Ccr for telephonyGroupFK.
6.506	Added recordingTones column to the VoIPMonitorDevice table.
6.505	Created the ReconciliationHistory table.
6.504	Added immediatedScreenUpload column to Workflow table.
6.503	Added Index for EvalQuestion table.
6.501	Added displayOrder to the MetaDataField table.
6.498	Updated the servermaxConcurrentRecordings to 1000 instead of 100 now that screen recording is on the client desktop.
6.497	Removed legacy UniqueCcr table.
6.496	Assure that there is an UseSSL flag for all Screen and Voice Servers.
6.495	Created cascade for person and record ID in ReportUserConfig.
6.491	Created and populated the CdrCcr and indexes.
6.486	Changed the Current Best Performers Widget to use a single value for bands rather than a range, dropped band5.
6.480	Added support for point based Evaluations. Created EvalFormScoringType, EvalFormQuestionOptionType, EvalFormQuestionOption, EvalFormQuestionTemplate, and EvalFormQuestionTemplateOption tables.
6.442	Added versionInfo (client version) column to LoginState table.
6.441	Added types to the CcrType table.
6.437	Dropped telephoneNumber from VoipMonitorDevice.
6.436	Created VoipMonitorDeviceLine table to allow multiple line configuration of devices.
6.434	Added flags to RecordingEvent table for if an edit of the audio is required and if an edit of screen is required.
6.431	Added a bit column to WorkflowClassifier to flag if screen should be uploaded.

Schema Version	Description
6.430	Added StartTime to the Screen table.
6.424	Added telpehoneNumber column to VoipMonitorDevice table.
6.429	Changing the Workflow Classifier isInbound column to be that of the newly created table WorkflowClassifierNumberType.
6.423	Added email related fields to CCR table.
6.420	Added isPlayable and mimeType to FileType table.
6.418	Created ScreenMonitoring table.
6.417	Updates to CCR indexes.
6.409	Added team and group keys to CCR table.
6.401	Added original and destination conversation keys to CCR table.
6.399	Created default population of LoginState table columns.
6.393	Changed CCR trigger and constraint.
6.385	Created the RecordingApiCommand table.
6.376	RecordingEventType id is now auto-increment.
6.364	Added server property to store active directory administration user group.
6.361	Added calling and called party number information to CDR.
6.353	Merged voice and screen IP address RecordingStateAudit information into a single column.
6.337	Created table for AXCL UCCM Schema mapping.
6.335	Added tables for Gateway Recording Reconciliation.
6.327	Modified EvalComment for editable comments
6.324	Added tables for the ACD filter.
6.285	Added indexes on Ccr and Metadata.
6.254	Modified upload states and reason codes in Ccr.
6.244	Modified Audio/Screen tables for inserting recordings at the start of the call rather than at the end of a call.

Schema Version	Description
6.230	Added tables for Workflow processing.
6.209	Added tables for Recording Clusters.
6.199	Added tables for Scheduled Reports.

Database Change Log for Schema 5

Schema Version	Description
5.580	Added/modified indexes for performance.
5.576	Added LoginState and RealTimeRecordingMonitorState tables for use with Real Time Recording Monitoring application.
5.550	Added configuration for MediaSense.
5.546	Renamed Search table to Filter and added column so it can be shared with multiple applications.
5.528	Added Site table and foreign keys to allow multiple sites.
5.513	Added Ccr.calibrationFormFK as a reference to the calibration form which could be different from a form that is used for a counted score.
5.507	Removed License table and references.
5.504	Added a DashWidgetSetting.isLockable column to determine whether or not a column can be locked down by an administrator.
5.500	Added hidden Administrator role and adds new person with administrator role.
5.492	Added the following items for calibration: • Added the Ccr.isCalibration flag and removed constraints to allow multiple evaluations per contact. • Added the isCountedScore, evalStateFK and evaluatorFK to Eval table. Backward Compatibility Warning: If JOINing on Eval from Ccr, users should add Eval.isCountedScore=1 to filter out calibration scores if those evaluations are not to be counted.

Schema Version	Description
5.483	Added role-based defaults to dashboard. New association table: DashRoleView and added isAdminCreated column to DashView.
5.480	Added Ccr.ccrTypeFk to specify whether the contact is a call or non-call contact.
5.476	For non-call contacts, allow call-based columns in Ccr to be nullable. Columns are: ani, dnis, line, icmCallId, callDuration, and associatedCallId.
5.466	Added order for displaying apps in Unified Workforce Optimization.
5.460	Changed Ccr.associatedCallId size and type to VARCHAR(52).
5.459	Add signalingMethod column to the TelephonyGroup. Flag for whether it is CTI or MediaSense based recording.
5.458	Renamed VoipMonitorRecordingType table to RecordingType and added Ccr.recordingTypeFK to refer to how this contact was recorded.
5.453	Changed size and type of Ccr.icmCallid column to VARCHAR (128).
5.448	Added loggedInPersonFK column. this column shows who is currently logged in to the device. It is used for Hot Desking and Extension Mobility purposes. For explicitly assigned agents, use 'personFK'.
5.170	Added Server, ServerType, and ServerProperties table to store Server configuration.
5.173	Added VoiPMonitorDevice and Gateway tables to store SPAN recording configuration.
5.175	Added qualityWorkflowFK and archiveWorkflowFK to Team table to refer to Workflow configurations.
5.177	Added LicenseLibrary table to hold license information. The data in this table is encrypted (we do not publish the encryption key).
5.181	Added workflowXML and name columns to Workflow table to store workflow information that was previously stored only in LDAP.

Schema Version	Description
5.215	Moved Voice/Screen server information form Host table to new Server tables (see 5.170), changed hostFK to serverFK references in Audio/Screen tables, and dropped the Host table.
5.226	Changed Ccr.agentFK to Ccr.personFK and updated indexes/-constraints. This was done for performance and usability purposes—a person could have multiple Agent records (f their agent role was modified) but only one Person record.
5.234	Moved displayId from Agent to Person table. Added isArchiveUser to Person.
5.235	Dropped EvalComment.rol column (comments are made by a person, not a role).
5.242	Added isConfigured column to Person. Changed username/domainName to be nullable – Previously, only Persons configured for QM were stored in the Person table (unconfigured people were in LDAP).
5.253	Added isManager, isSupervisor to Person table (for performance purposes). Changed size of names, displayId and skillTargetId in Person.
5.268	Modified EvalForm and EvalFormQuestion tables for question weighting features.
5.276	Added Role, Alert, and Eval Alerts table for alert feature.
5.281	Modified VoiPMonitorDevice table to add server references.
5.283	Added acdFirstName and acdLastName to Person table for names coming from the ACD instead of AD.
5.291	Modified VoiPMonitorDevice table for changes to server references.
5.294	Added Workflow.canChangeForm column for ability for allowing evaluators to change form feature.
5.295	Changed data type for Eval.scorePerscent form int to float.
5.298	Added Alert table to store user alerts.
5.300	Added RtpFilter table to store port and IP information that could be filtered from recording (this is used to filter out CAD recording/monitoring traffic.

Schema Version	Description
5.318	Added ReportType, Report, ReportParameter, ReportRequiredParam, ReportRole, ReportRoleScope, ReportColumn, ReportUserConfig, ReportConfiguration tables. These tables were moved from the SQMREPORTDB catalog that was dropped in 8.0. Some tables are new to support the new report features.
5.324	Removed Person.isLicensed column. This information was redundant to the licensing API used by QM. This was dropped to avoid confusion.
5.329	Added "file is cleaned" upload state and changed existing Ccr data for files that have been cleaned. This was done for performance reasons.
5.335	Added fileSize to Audio/Screen table to track the size of the recording files.
5.341	Removed unnecessary Locale table—only support one locale at this time.
5.342	Changed Person.skillTargetId column to be case sensitive.
5.369	Added RecordingEventType and RecordingEvent tables to track call events (talk over and silence).
5.372	Added UI_Component, UI_FieldType, and UI_Field tables to allow configurable column in the UI.
5.386	Added License, UI_FeatureGroup, UI_Feature, and UI_FeaturePermissions tables to track what roles can see what widgets (for example, Apps menu).
5.390	Increased size of DbProperties.setting column to be 1024 characters.
5.391	Removed EvalForm.displayFlags column. This is replaced by the configurable columns feature added previously (UI_Field, etc).
5.392	Added UI_FeatureGroup, UI_Feature, and UI_FeaturePermissions tables to handle application menu permissions.
5.396	Removed displayFlags column for EvalForm that stored column permissions per evaluation form—changed to permissions per user.

Schema Version	Description
5.397	Added isHotdeskDefaultUser column in Person. Indicates whether or not this is the default user for hot desk recording.
5.398	Modified application menu tables–UI_Feature.
5.404	Added tables for Dashboard Widgets and Views.
5.410	Added the following configurable column tables: Field, FieldCategory, and PersonField.
5.438	Added Search table for saved searches.
5.441	Added the audio offset for comment called EvalComment.audioRecordingOffset.
5.444	Removed the section comments column called EvalComment.evalFormSectionFK.