

Cisco Integration

Integration Guide

QMS 9.4

This document contains information about the configuration of Quality Management Suite for recording Cisco calls.



Table of Contents

About This Document	3
Solution Overview	4
SIP and SCCP Port Mirroring	5
PBX Configuration	5
Network Configuration	5
QMS Configuration	6
Forked Audio – IP Phone-based Recording	7
PBX Configuration	7
TSP Installation and Configuration	9
QMS Configuration	10
Forked Audio – Network-based recording	11
PBX Configuration	11
TSP Configuration	12
QMS Configuration	12
Media Encryption	13
PBX Configuration	13
TSP Configuration	15
QMS Configuration	16
QMS High Availability and Forked Audio	19
PBX Configuration	19
QMS Configuration	21
QMS Cisco Phone Controls	22
IIS Configuration	22
PBX Configuration	23
Assigning Cisco Phone Controls to an IP Phone's Hard Key (optional)	25
CUCM User Options Configuration (optional)	26
QMS Configuration File Settings (Optional)	27
Troubleshooting	29
Terms of use	32
End User License Agreement	33

About This Document

Audience

This document is for the contact center system administrator who installs, configures, and integrates Cisco Unified Communications Manager (CUCM) with Enghouse Interactive Quality Management Suite (QMS).

Reference materials

The content contained in this document works in combination with the information and procedures in the following documents or Help.

- Quality Management Suite System Design Guide
- Quality Management Suite Installation Guide
- Quality Management Suite High Availability Guide

Document conventions

This document uses the following text formats and notation conventions.

Text format

Bold text indicates a button, field, link, option name, or similar function requiring an action.

Italicized text indicates new terms, directory paths, or references to external documents.

Notes and cautions

Icons used throughout this document identify additional details or special conditions.



Note

Provides additional information or describes special circumstances.



Caution

Warns of user actions that may cause system failure or irreversible conditions.



Stop

Describes actions that you should only perform under the supervision of Enghouse Customer Support.

Contact information

For more information regarding Enghouse products, services, and support, please visit www.enghouse.com.

Solution Overview

The QMS Call Recording solution supports two general methods of recording calls for Cisco UCM, with the second method having two variants. The first method of recording is accomplished through the use of packet capturing (also called packet sniffing) on a NIC installed in the QMS server. The network switch(es) that Cisco IP phones are connected to must have port mirroring enabled, in order to forward RTP audio packets and SIP or SCCP call control packets to the QMS Call Recording Service.

The second method, known as Forked Audio, forks (copies) the inbound and outbound audio streams to the QMS Call Recording Service over a SIP trunk. There are two types of Forked Audio, IP Phone-based recording and Network-based recording. IP Phone-based recording is available for Cisco IP phones that have Built-In Bridge (BIB) capability. The BIB is responsible for forking the audio streams in order to send a copy of the audio to the QMS Call Recording Service. Network-based recording, on the other hand, relies on a separate network device to fork the audio, rather than the IP phones themselves. Both flavors of forked audio recording require the Cisco Telephony Services Provider (TSP) to be installed on the QMS server in order to receive call control events from CUCM.



Cisco UCM supports the ILBC, ISAC, and Opus audio codecs which employ the use of a “dynamic” RTP Payload Type. Dynamic payload types have a value in the range of 96 to 127 and the value used for a specific codec can vary. To process audio encoded using ILBC and ISAC, QMS must know which dynamic payload type the Cisco UCM is employing. The following settings may be used to provide that information.

IlbcCodecPayloadTypes: Cisco audio may be encoded using the ILBC codec. QMS looks for ILBC to be sent in RTP packets with the (dynamic) PayloadType value of 116. If your UCM employs a different value, it should be specified via the `IlbcCodecPayloadTypes` setting in the `CallRex\System` section of both the `ConsolidatedRecordingService.exe.config` and `ManagedCRAC.exe.config` files.

IsacCodecPayloadTypes: Cisco audio may be encoded using the ISAC codec. QMS looks for ISAC to be sent in RTP packets with the (dynamic) PayloadType value of 124. If your UCM employs a different value, it should be specified via the `IsacCodecPayloadTypes` setting in the `CallRex\System` section of both the `ConsolidatedRecordingService.exe.config` and `ManagedCRAC.exe.config` files.

OpusCodecPayloadTypes: Cisco audio may be encoded using the Opus codec. QMS looks for Opus to be sent in RTP packets with the (dynamic) PayloadType value of 114. If your UCM employs a different value, it should be specified via the `OpusCodecPayloadTypes` setting in the `CallRex\System` section of both the `ConsolidatedRecordingService.exe.config` and `ManagedCRAC.exe.config` files.

If your UCM employs a PayloadType value of 114, you will need to specify a different value for the RTC16 codec in the `ManagedCRAC.exe.config` file by setting the `RTC16CodecPayloadTypes` to an unused dynamic RTP Payload Type. For example, `RTC16CodecPayloadTypes="110"` if payload type 110 is not used in your environment.

SIP and SCCP Port Mirroring

Port mirroring is the action of copying all inbound and outbound network traffic from one port on a network switch to another port. Most modern network switches allow for some level of port mirroring, including Virtual LAN (VLAN) configurations that allow the network administrator to segment the local network as needed. QMS can utilize port mirroring to record calls to and from Cisco IP phones. Cisco uses either the SIP protocol or the SCCP (Skinny) protocol to send call control information. QMS has the capability of handling either protocol, or both protocols at the same time.

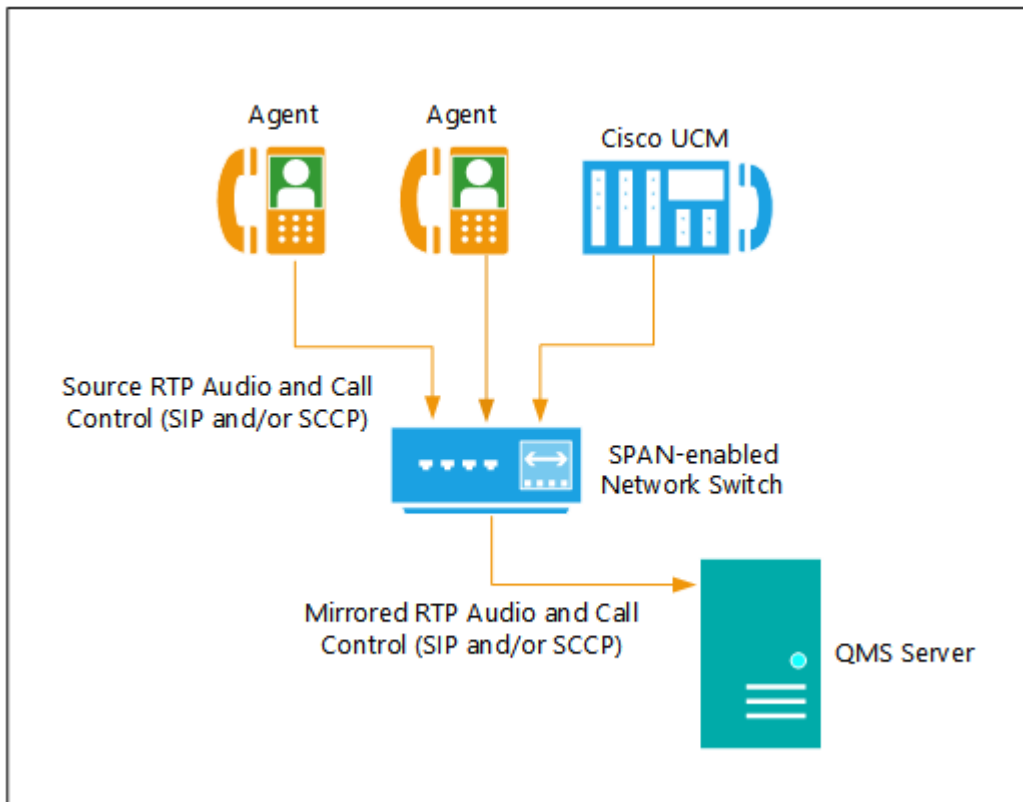


Figure 1: Port mirroring configuration

PBX Configuration

There are no changes required in CUCM for port mirroring integrations.

Network Configuration

While different brands and models of network switches have unique methods of configuration, setting up network mirroring usually involves accessing the network port settings through the network switch's administration portal. One port is selected to be the destination port for the mirrored packets; this port might be selected by default, or the port might be selectable by the administrator. The destination port is the port to which the QMS server's capture network interface card (NIC) is connected. After setting up the destination port, the source ports from which network traffic will be mirrored must have mirroring enabled. The source ports will be ports to which the Cisco IP phones are connected. Refer to the network switch's administration manual for more specific details on configuring port mirroring for the switch.

QMS Configuration

Services configuration:

1. Log into the QMS Client as a user with permissions to create new or update existing Services information.
2. Click on **Services** under Administration in the navigation area.
3. Select the Call Recording Service that will have users assigned, and click **Edit**.
4. A new tab will appear, displaying the settings for the service. The PBX Type should default to SIP – if the Cisco environment is using SIP devices, leave the PBX Type as SIP (or select SIP from the drop-down if it isn't already selected). If the Cisco environment is using SCCP (Skinny), select the option titled "Cisco SCCP" from the drop-down. If the Cisco environment is using a mix of SIP and SCCP (Skinny), select the option titled "Cisco SCCP+SIP" from the drop-down.
5. Enter in the **PBX IP Address**.
6. Enter the **Address override** and **Secondary Endpoint Subnet** fields if needed.
7. Click **Save**.

User configuration:

1. Log into the QMS Client as a user with permissions to create new or update existing User information
2. Click on **Users** under Administration in the navigation area.
3. Click **Add** to create a new user, or if users already exist in the system, select a user to update from the user list and click **Edit**.
4. Fill out the **Personal** and **Account Information** sections.
5. Check the **Call Recording** checkbox in the Licenses section.
6. In the Call Recording Settings section, pick the name of the server used to record the user.
7. Fill in the **Extension** and **Endpoint IP Address** fields accordingly.
8. If the service's PBX Type is set to "SIP", fill in the **Additional Extensions** field, if necessary.
9. If the service's PBX Type is set to "Cisco", indicating an SCCP environment, fill in the **Private Extensions** field if necessary.
10. Click **Save**, and repeat for each additional QMS user.

Forked Audio – IP Phone-based Recording

Forked Audio – IP phone-based recording refers to a type of recording where audio is delivered to the QMS server via interactions between the IP phones, CUCM, the Cisco Telephony Service Provider (TSP) installed on the QMS server, and QMS itself, rather than via port mirroring. This type of recording is available only on phone devices that have Built-In Bridge functionality. The Built-In-Bridge (BIB) will fork (copy) both inbound and outbound RTP audio streams to the QMS server over a SIP trunk to allow the call to be recorded. Call Control is sent to the QMS server from CUCM via TAPI through the TSP.



The G.722 Codec must be enabled on the Call Manager under the 'Clusterwide Parameters (System - Location and Region)' settings.

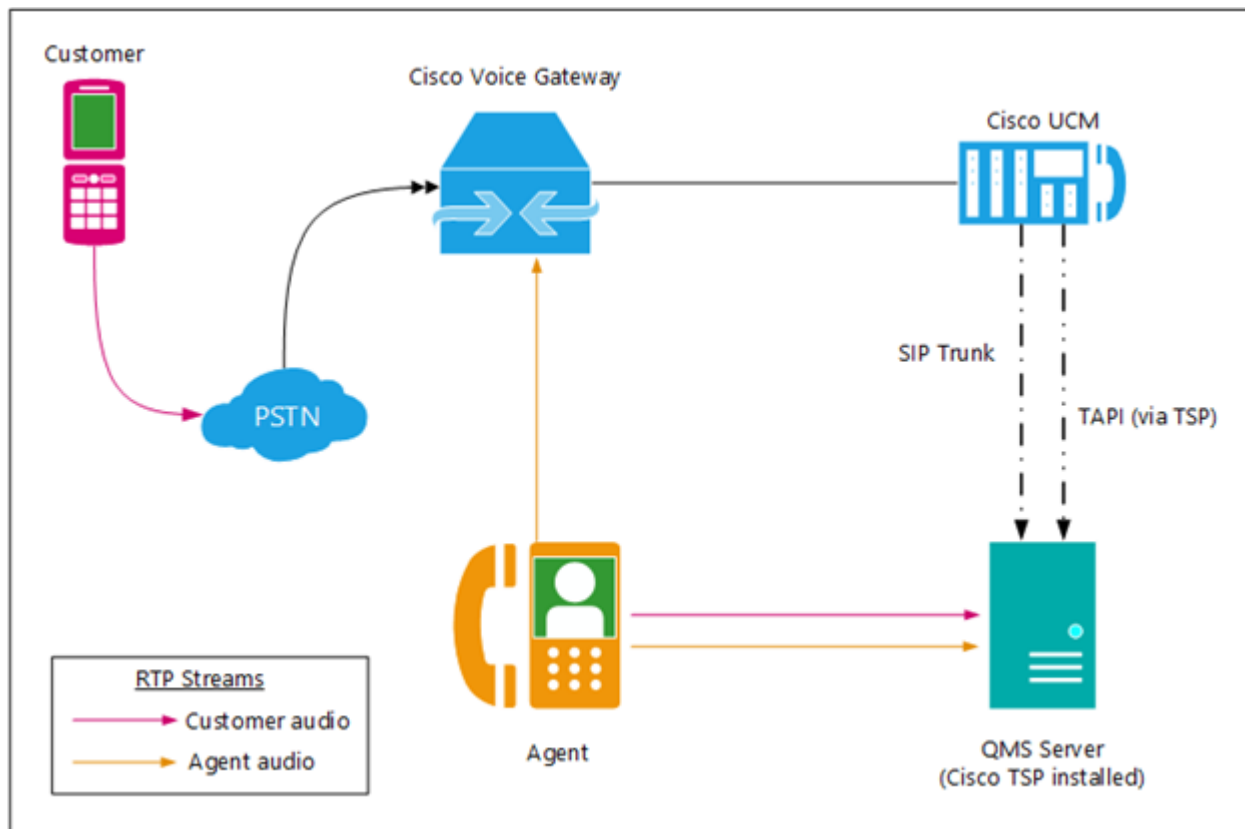


Figure 2: Forked Audio – IP phone-based recording configuration

PBX Configuration



The steps listed below may be slightly different, based on the version of Cisco UCM installed.

1. **SIP Profile:** At least one SIP Profile should already exist within CUCM, usually named **Standard SIP Profile**. Unless special SIP Profile settings are needed in the environment, the Standard SIP Profile should be sufficient for connections to the QMS server.
2. **SIP Trunk Security Profile:** At least one SIP Trunk Security Profile should already exist in CUCM, usually named **Non Secure SIP Trunk Profile**. This Profile's default settings are unencrypted and unauthenticated, incoming protocol set to TCP+UDP, and outgoing protocol set to TCP. The incoming SIP port is set to 5060. These settings are usually sufficient for a general connection to the QMS server. A new Profile can be created, if needed, for example to create a secured connection to the QMS server (see the Media Encryption section below for more information on setting up a secure connection).
3. **Recording Profile:** At least one Recording Profile must be created in order to provision line appearances on agent phones for call recording. A Recording Profile specifies the destination address of the QMS server, either by a directory number, which could be a route pattern, or by the URL of the QMS server. To create a Recording Profile:
 - a. Log into the CUCM Administration Site with a user who has permission to create Recording Profiles.
 - b. Click on the **Device** menu
 - c. Click **Device Settings**
 - d. Click **Recording Profile**.
 - e. Click **Add New** to bring up the Recording Profile Configuration view.
 - f. Enter a **Name** and a **Recording Destination Address** for the profile.
 - g. Click **Save**.
4. **SIP Trunk:** In a Forked Audio configuration, a SIP trunk to the QMS server is needed to send the RTP audio streams to QMS for call recording. To set up a SIP trunk:
 - a. Log into the CUCM Administration Site with a user who has permission to create SIP trunks.
 - b. Go to the **Device** menu, then select **Trunk**.
 - c. Click **Add New**.
 - d. Select **SIP Trunk** on the Trunk Type drop-down list.
 - e. Leave the Device Protocol and Trunk Service Type fields set to their default values, and click **Next**.
 - f. On the Trunk Configuration view, give the trunk a **Device Name** in the Device Information section.
 - g. For QMS purposes, the default values in the IME, MLPP, and Call Routing Information sections should be acceptable.
 - h. In the SIP Information section, specify the **Destination Address** (IP Address of the QMS server) and **Destination Port** values in the Destination sub-section. The Destination Port value must match the **Incoming Port** value of the SIP Trunk Security Profile specified in the next step below.
 - i. Select the appropriate **SIP Profile** and **SIP Trunk Security Profile** settings, from the respective drop-downs, either to the default Profiles or to Profiles that were created in sections 2 and 3 above.
 - j. Click **Save**.
5. **IP Phones:** Adding IP phones to CUCM is beyond the scope of this document. Please refer to appropriate Cisco documentation for steps on adding IP phones to CUCM. To configure a phone for recording, do the following:
 - a. Log into the CUCM Administration site with a user who has permission to update Phone settings.
 - b. Go to the **Device** menu, click **Phone**, then click a phone's **Device Name** on the Find and List Phones view.

- c. In the Device Information section of the Phone Configuration view, set the **Built in Bridge** field to **On**. This setting will cause the phone to send a copy of the inbound and outbound RTP audio streams to the QMS server for recording.
 - d. Click **Save**.
 - e. In the Association Information section, click on the **Line [1]** link to access the Directory Number Configuration view.
 - f. Scroll down to the **Line 1 on Device...**section.
 - g. Set the Recording Option field to **Automatic Call Recording Enabled**.
 - h. Set the Recording Profile to the profile created in step 3 above.
 - i. Click **Save**.
 - j. For any other line appearances that should be recorded, repeat steps e to i for each line.
 - k. Repeat for each IP phone that should be recorded.
6. **Application User:** A CUCM Application User refers to the user assigned to the Cisco Telephony Service Provider (TSP) application, which is installed on the QMS server, whose credentials will be used to create the TAPI connection between CUCM and QMS via the TSP. This connection will transfer call control events from CUCM to the QMS server. To create an Application User:
- a. Log into the CUCM Administration site with a user who has permission to create Application Users.
 - b. Click on the **User Management** menu, then click **Application User**.
 - c. Click **Add New** to access the Application User Configuration view.
 - d. Enter in a **User ID**, then a **Password** (if desired), and confirm the Password.
 - e. In the **Device Information** section, select the phones to be recorded from the **Available Devices** list, and click the down arrow to move those phones to the **Controlled Devices** list.
 - f. In the Permissions Information section, click **Add to Access Control Group** to access the Find and List Access Control Groups window.
 - g. Click **Find** to bring up the full list of Access Control Groups.
 - h. Mark the checkboxes for the following groups:
 - Standard CTI Allow Call Monitoring
 - Standard CTI Allow Call Park Monitoring
 - Standard CTI Allow Call Recording
 - Standard CTI Allow Control of All Devices
 - Standard CTI Allow Control of Phones supporting Connected Xfer and conf
 - Standard CTI Enabled
 - i. Click **Save**.

TSP Installation and Configuration

1. Log into the Cisco UCM Administration site from the QMS server.
2. Click on the **Application** menu, then click on **Plugins**.
3. Click **Find** to pull up the list of all available plugins.
4. Scroll down the list, find the **Cisco TAPI 64-bit Client**, and click the **Download** link for that plugin.
5. Once the download is completed, start the installer (CiscoTSPx64.exe).

6. On the Choose Destination Location view, specify 1 for the number of TSPs to install, then click **Browse** to select a new Destination Folder if the default location is not desired.
7. Click **Next** to access the Configure TSP Instance view.
8. Enter in the **User ID** of the Application User created in section 7 above, along with a **Password** if one was entered on the Application User settings.
9. In the **CTI Manager 1** field, enter in either the IP address, IPv6 address, or hostname of the CUCM server.
10. Select the **address type** from the drop-down, based on the type of information entered in the CTI Manager 1 field.
11. Click **Next** to access the Media Driver/Auto-Upgrade/TFTP Server IP Address view.
12. In the Cisco Media Driver Configuration section, change the port numbers if the default range specified would conflict with other applications or services on the network.
13. Set the **Auto-Upgrade Option** to the desired setting.
14. Mark the checkbox for **Start Cisco TSP Notifier when Windows Starts**, if not already checked.
15. Click **Next** to finish the installation, then reboot the QMS server when prompted.

QMS Configuration

Services Configuration

1. Log into the QMS Client as a user with permissions to create new or update existing Services information
2. Click on **Services** under Administration in the navigation area.
3. Select the Call Recording Service that will record the Forked Audio phones from the list of services, and click **Edit**.
4. Set the PBX Type to **Cisco Forked Audio**
5. Enter the IP address of the CUCM server in the PBX IP Address field
6. Click **Save**.

User Configuration

1. Log into the QMS Client as a user with permissions to create new or update existing User information.
2. Click on **Users** under Administration in the navigation area.
3. Click **Add** to create a new user, or if users already exist in the system, select a user to update from the user list and click **Edit**.
4. Fill out the Personal and Account Information sections.
5. Check the **Call Recording** check box in the Licenses section.
6. In the Call Recording Settings section, pick the name of the server used to record the user.
7. Enter the Primary Extension, Endpoint IP Address (i.e. the IP address of the phone), and any additional extensions (line appearances) the user may have.
8. Click **Save**, and repeat for each additional QMS user.

Forked Audio – Network-based recording

Network-based recording is very similar to IP phone-based recording; the inbound and outbound RTP audio streams are forked (copied), with the duplicate streams being sent to the QMS server for recording a call. The difference between the two types of forked audio recordings is the device that does the forking of the audio streams. With network-based recording, the IP phones do not fork the audio; rather, another device registered with CUCM does the forking of the audio streams. This device could be a Cisco Voice Gateway, a Cisco Unified Border Element (CUBE) device, a Media Termination Point (MTP), or a Trusted Relay Point (TRP). From the QMS server's perspective, all of the audio is coming from one source IP address. In order to distinguish which audio streams belong to which agent, using just the IP address will not work in this scenario. Therefore, each agent will be assigned a port, such that the IP address/port combination will be unique per agent. QMS will then be able to create call recordings for each agent that contains the correct audio streams.

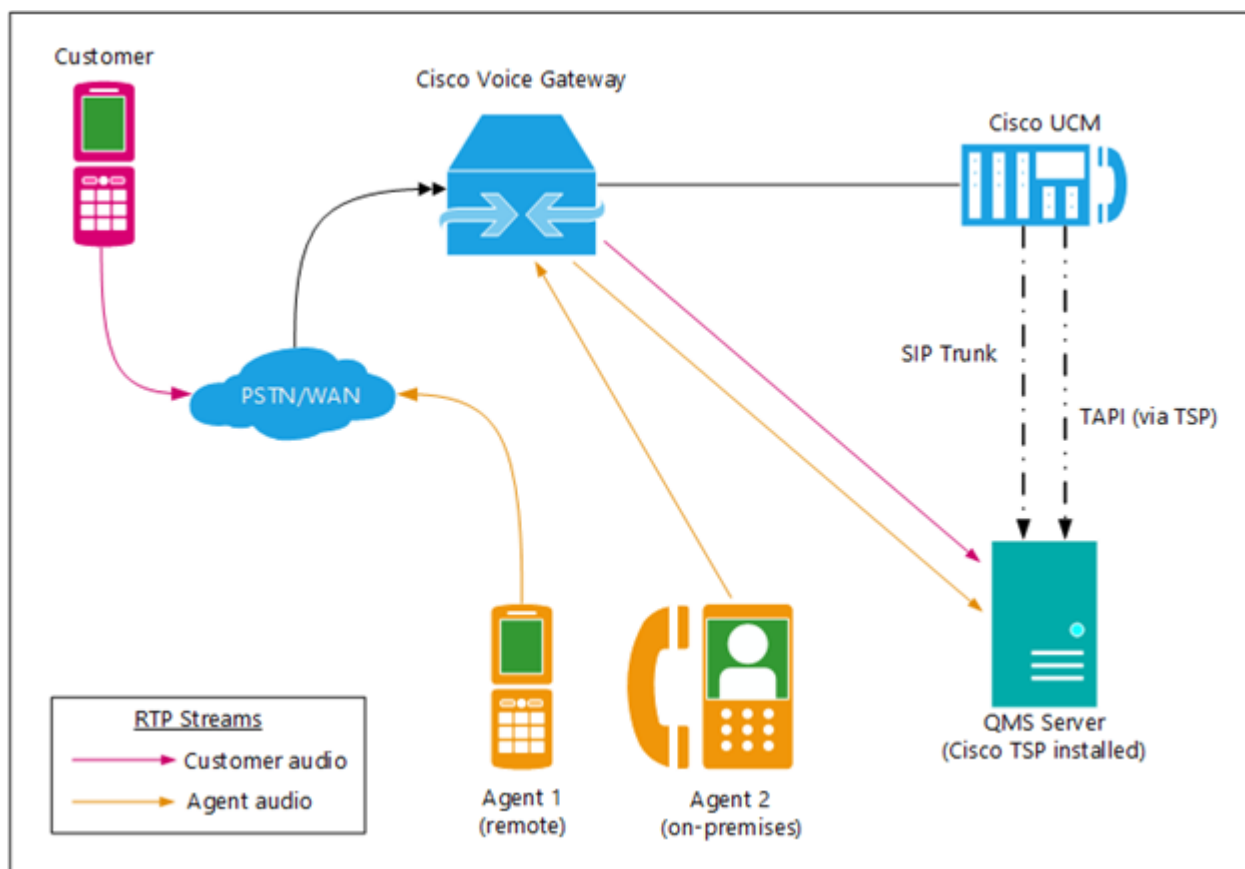


Figure 3: Forked Audio – Network-based recording configuration

PBX Configuration

1. Configuration of the various network-based recording devices within CUCM is beyond the scope of this document. Please refer to the appropriate Cisco documentation regarding setup and configuration of Voice Gateways, CUBE devices, MTPs, or TRPs as necessary.

2. The steps outlined in the PBX Configuration section under Forked Audio – IP Phone-based Recording above also apply to network-based recording. Follow the steps outlined in that section to set up the necessary profiles, application (TSP) user, SIP trunk, and phone settings.

TSP Configuration

1. The steps outlined in the TSP Installation and Configuration section under Forked Audio – IP Phone-based Recording above also apply to network-based recording.. Follow the steps outlined in that section to set up the TSP for network-based recording.

QMS Configuration

1. The steps outlined in the QMS Configuration under Forked Audio – IP Phone-based Recording above also apply to network-based recording. Follow the steps outlined in that section to set up the QMS services and users. After those steps have been completed, the following settings need to be added to the CallRecordingService.exe.config file, located in the QMS installation folder, in the <Recording /> section. The key names are case-sensitive.
2. **DisableCiscoFADynamicAudioPorts:** The configuration setting that turns on/off the use of dynamic ports for Forked Audio recording. Setting this value to true will be used for IP phone-based recording, where audio is being forked at the phone level, meaning the audio streams will be sourced from the IP address of each user's phone. Setting this config value to false turns on dynamic ports, allowing network-based recording where all audio streams come from one IP address.
3. **DisableCiscoFADynamicAudioPortUserPersistence:** This configuration setting specifies whether associations between users and unique ports are maintained when a user is idle. If this value is set to true, when a user moves into an idle state, the ports assigned to that user are returned to the pool of available ports. If this value is set to false, which is the default, the user/port associations are maintained regardless of user state.
4. **CiscoFADynamicAudioPortBase:** This configuration setting sets the first audio port in a range of ports to be used in network-based recording. The default audio port base is port 16000.
5. **CiscoFADynamicAudioPortRange:** This configuration setting sets the number of ports available for dynamic port assignment in a network-based recording scenario. With allowances for RTP Control Packet (RTCP) traffic, the minimum number of ports per user is 8, while the maximum number of ports per user is 10, which allows for more intuitive interpretation of port usage when reviewing logs. The default port range is 2000, which will accommodate persistent port assignment for 200 users.
6. NOTE: The default values for CiscoFADynamicAudioPortBase and CiscoFADynamicAudioPortRange specify a default port range of 16000 – 17999.

Media Encryption

Media encryption, also called secure media, refers to configuring CUCM to allow devices to register as secure devices, if they are capable. For instance, if an IP phone has secure media capability, it will register with CUCM as a secure device. Audio for calls to/from that phone will then be encrypted, if the far-end device is also a secured device.

In order to enable encrypted audio in a CUCM environment, there are several steps that need to be completed, both on the PBX end and on the TSP end. On the PBX, the overall security mode needs to be updated, followed by creating secure phone profiles and secure SIP trunk profiles. Phones need to be updated, and trunks need to be created/updated to use the secure profiles. Lastly, Application User Certification Authority Proxy Function (CAPF) Profiles need to be created for each Application User that is to be used to establish a secure connection between PBX and TSP. The Application Users must then be updated to use the new CAPF Profiles. On the TSP side, a secure Application User must be entered, and the Security settings need to be updated.

Provided in this section is a condensed version of the process needed to enable encrypted audio in a Cisco UCM environment, in preparation to use encrypted audio in a QMS installation that uses Forked Audio. While these steps will cause audio and call control to be encrypted in general, QMS only supports encrypted audio in a Forked Audio installation. Full details for the steps listed in the PBX Configuration section below can be found in the Cisco Unified Communications Manager Security Guide.

PBX Configuration

1. Set the Cluster Security Mode to Mixed Mode on the PBX
 - a. Purchase at least two USB security keys from Cisco.
 - b. From a PC or server that is able to communicate with the PBX, download the CTL Client.
 - i. Log into the CUCM Admin site.
 - ii. Go to Application > Plugins, and click **Find**.
 - iii. Find the CTL Client in the list, click the **Download** link, and run the installer.
 - c. Launch the CTL Client after the installation has completed.
 - d. Select the option **Set Cisco Unified CallManager Cluster to Mixed Mode** and click **Next**.
 - e. Insert the Cisco security keys when prompted.
 - f. After both keys have been processed and added to the PBX, click **Finish**.
 - g. Reboot the PBX when prompted and close the CTL Client.
2. Set up secure phone profiles
 - a. Log into the CUCM Administration site with a user that has admin-level permissions.
 - c. Go to **System > Security > Phone Security Profile**. Create a new Phone Security Profile with the following settings:
 - i. **Device Security Mode** = Encrypted.
 - ii. **TFTP Encrypted Config** enabled.
 - iii. **Authentication Mode** = By Existing Certificate (precedence to MIC).
 - iv. **Key Size** = 1024 (default, 2048 can be used if stronger encryption is desired).
 - c. Click **Save**. No phones have been assigned to the profile yet, so Apply Config is not necessary.
3. Set up an Encrypted SIP Trunk Security Profile
 - a. Log into the CUCM Administration site with a user that has admin-level permissions.

- b. Go to **System > Security > SIP Trunk Security Profile**.
 - c. Create a new SIP Trunk Security Profile with the following settings:
 - i. **Device Security Mode** = Encrypted.
 - ii. **Incoming Transport Type** = TLS.
 - iii. **Outgoing Transport Type** = TLS.
 - iv. **Incoming Port** = 5061.
 - d. Click **Save**. No SIP Trunks have been assigned to the profile yet, so Apply Config or Reset are not necessary.
 4. Configure phones to register with the PBX as Encrypted
 - a. Log into the CUCM Administration site with a user that has admin-level permissions.
 - b. Go to **Device > Phone**, click **Find**.
 - c. Click the device name for the phone.
 - d. Scroll down the list of phone settings and find **Device Security Profile**.
 - e. Set the **Device Security Profile** to the Secure Profile for the given phone model.
 - f. Click **Save**, then Click **Apply Config**.
 - g. If the phone doesn't reset automatically, then click **Reset**.
 - h. Verify the phone registers with the PBX as Encrypted by doing the following:
 - i. Press the settings button on the phone.
 - ii. Scroll down the menu and select **Security Configuration**.
 - iii. Security Mode will say **Encrypted** if the phone registers securely with the PBX, and **Non-Secure** otherwise.
 5. Create encrypted SIP trunks for each TSP that needs to have a secure connection to the PBX.
 - a. Log into the CUCM Administration site with a user that has admin-level permissions.
 - b. Go to Device > Phone, click **Find** and select an existing trunk or click **Add New** to create a new trunk. Set the following:
 - i. **Enable SRTP** Allowed.
 - ii. **Consider Traffic on This Trunk Secure** = When using both sRTP and TLS.
 - iii. **SIP Destination Port** = 5061.
 - iv. **SIP Trunk Security Profile** = (the encrypted SIP Trunk profile created above).
 - c. Click **Save**. If prompted to do a reset, click **Reset**.
 6. Create Application (TSP) User CAPF Profiles to establish secure communication with the PBX
 - a. Log into the CUCM Administration site with a user that has admin-level permissions.
 - b. Go to User Management > User Settings > Application User CAPF Profile and click **Add New**. Set the following:
 - i. Select an **Application User** from the list that will be used to establish a secure connection between PBX and TSP.
 - ii. Enter an **Instance ID**, which can be any valid string of text.
 - iii. **Certificate Operation**: Install/Upgrade.
 - iv. **Authentication Mode**: By Authentication String.

- v. Click the **Generate String** button to populate the Authentication String.
 - vi. Set the desired **Key Size**.
 - vii. Set the **Operation Completed By** to a date far enough into the future to allow the TSP to be installed and configured.
- c. Click **Save**.
7. Create an Application (TSP) User for each TSP requiring secure communication with the PBX.
- a. Log into the CUCM Administration site with a user that has admin-level permissions.
 - b. Go to **User Management > Application User**, click **Find** and select an existing Application User, or click **Add New** to create a new Application User.
 - c. Enter a **User ID** and **Password**.
 - d. For the devices this Application User is supposed to control, move the devices from the **Available Devices** list down to the **Controlled Devices** list.
 - e. Confirm the **Associated CAPF Profiles** list contains the CAPF Profile created previously for this user.
 - f. Add the following three groups to the Groups list under **Permissions Information**:
 - i. Standard CTI Allow Reception of SRTP Key Material.
 - ii. Standard CTI Enabled.
 - iii. Standard CTI Secure Connection.
 - g. Verify Roles of the same names have been populated in the Roles list.
 - h. Click **Save**.

TSP Configuration

1. On the TSP side, configuration for encrypted media is contained on the Security tab of the Cisco TSP Configuration. On the QMS server, right click the Cisco TSP Notifier icon, and select Cisco **TAPI Configuration**.
2. Highlight **CiscoTSP001.tsp** (this is the default name of the configuration profile, and may have been named differently during TSP installation), then click **Configure**.
3. Click on the **Security** tab.
4. Check both **Secure Connection to CTIManager** and **Fetch Certificate** checkboxes.
5. In the CAPF Settings section, enter the **Authorization String** and **Instance Identifier** values created in step 6.2 in the PBX Configuration section above.
6. Enter in the **CAPF Server IP** (usually the IP of the CUCM server), **CAPF Port**, **Number of Retries** and **Retry Interval** for Certificate Fetch values.
7. In the **TFTP Settings** section, enter in the **TFTP Server IP address** (usually the IP of the CUCM server).
8. Click **OK** to save the changes.
9. For the TSP to actually fetch the certificate information from CUCM, an application must exercise the TSP to communicate with CUCM. Stop the QMS Call Recording Service if it is running, pause 20 seconds to make sure all ports used by QMS have been released, then restart QMS Call Recording Service.
10. Access the **Security** tab within the Cisco TSP Configuration.
11. If the certificate information was successfully received, the Fetch Certificate checkbox will be unchecked, Certificate STATUS will report **Available**, and the Authorization String field will be blank.

QMS Configuration

Creating self-signed certificates on the QMS server: To enable security between QMS and the Cisco PBX using self-signed certificates, we need to generate two certificates:

- A root certificate that represents the Certificate Authority.
- A certificate that represents the identity of the client or service.

If the environment has more than one Call Recording Server, each Call Recording Server will need to have the Cisco TSP installed. The steps below will therefore need to be executed on EACH Call Recording server, because each server will need to have its own unique certificate to create a secure connection to the PBX.

There are three command-line utilities that are used to create the certificates. The utilities can be found in the Program Files (x86)\Windows Kits\8.0\bin\x64 folder. An alternative location, depending on what version of .NET is installed, would be the Program Files (x86)\Windows Kits\10\bin\x64 folder. If neither of these locations exist on the QMS server, download and install either the Windows 8.1 or Windows 10 SDK onto the QMS server. This will install the three utilities, which are:

- **Makecert.exe:** creates X.509 certificates, as well as the following:
 - o Generates a public key and a private key.
 - o Associates the key pair with a host name.
 - o Binds the host name to the public key.
- **Cert2spc.exe:** creates a Software Publisher's Certificate (SPC) from an X.509 certificate.
- **Pvk2pfx.exe:** creates a PKCS12 file from a certificate's private key and an SPC file.

The steps to create the self-signed certificates are as follows:

1. On the QMS server, start a Command Line window, using the Run As Administrator option.
2. First, a self-signed Root CA certificate needs to be created. Type the following command:

```
C:\> makecert -sv <server name>_RootCA.pvk -r -n "CN=<server name or IP address>" -len 1024
<server name>_RootCA.cer
```

- a. Replace the <> bracketed items with values relevant to the environment.
 - b. **-sv** specifies the file that contains the private key. The private key will be used for signing certificates issued by this certificate authority. Makecert will ask for a password to protect the private key in the file.
 - c. **-r** indicates that the certificate will be a root certificate because it is self-signed.
 - d. **-n** specifies the common name for the root certificate authority; this should be the name of the server. The convention is to prefix the name with "CN=" where CN stands for "Common Name".
 - e. **-len** specifies the key length in bits (default is 2048)
 - f. The <server name>_RootCA.cer file will contain the public key only.
3. Next, a certificate will need to be created for the QMS Call Recording service. Type the following command:

```
C:\> makecert -ic <server name>_RootCA.cer -iv <server name>_RootCA.pvk -n "CN=<server name or IP address>" -sv <server name>_Cert.pvk -pe -sky exchange -len 1024 <server name>_Cert.cer
```

- a. Replace the <> bracketed items with values relevant to the environment.
- b. **-ic** specifies the name of the root certificate (.cer) file created in step 1.2 above.
- c. **-iv** specifies the name of the container file for the private key of the root certificate created in step 1.2 above.

- d. **-n** specifies the name for the certificate.
 - e. **-sv** specifies the key file for the certificate. This must be unique for each certificate created. If a name is reused, an error message is displayed.
 - f. **-pe** specifies that the private key is exportable and is included with the certificate. For message security this is required because the corresponding private key is needed.
 - g. **-sky** specifies what kind of key is being created. Using the exchange option enables the certificate to be used for signing and encrypting messages.
 - h. **-len** specifies the key length in bits (default is 2048), and should be the same value specified in step 1.2 above.
4. Finally, a PKCS12 (.pfx) file needs to be created. First, run the following command to create the intermediate SPC file:
- ```
C:\> cert2spc <server name>_Cert.cer <server name>_Cert.spc
```
- a. Replace the <> bracketed items with values relevant to the environment.
5. Next, run the following command to create the PKCS12 file based on the certificate and SPC files:
- ```
C:\> pvk2pfx -pvk <server name>_Cert.pvk -spc <server name>_Cert.spc -pi <password> -pfx <server name>_Cert.pfx
```
- a. Replace the <> bracketed items with values relevant to the environment.
 - b. **-pvk** specifies the name of the file containing the private key created in step 1.3 above.
 - c. **-spc** specifies the SPC file created in step 1.4 above.
 - d. **-pi** specifies the password to be used to secure the PKCS12 file.
 - e. **-pfx** specifies the name of the PKCS12 file to be created.
6. Repeat steps 2 through 5 for every Call Recording Server

Import certificates into the QMS server's certificate store: On the QMS end, the PKCS12 and root certificates will need to be installed into the QMS server's certificate store. If there is more than one Call Recording Server in the environment, these steps will need to be run on EACH server. The following steps outline installing both certificates:

1. Start the Microsoft Management Console by right-clicking the **Start** button, selecting **Run**, typing in "mmc", and clicking **OK**.
2. In the MMC window, go to the File > Add/Remove Snap In menu item.
3. Under Available snap-ins, select **Certificates**, click **Add** to move it under **Selected** snap-ins.
4. In the "**Certificates snap-in**" window, select "**Computer account**" and click **Finish**.
5. On the Select Computer window, verify **Local Computer** is selected, and click **Finish**.
6. Click **OK**. Certificates should now appear under Console Root in the MMC window.
7. Expand the **Certificates** node.
8. To import the RootCA certificate, right-click on the **Trusted Root Certification Authorities** node, mouse-over **All Tasks**, and select **Import**.
9. On the Certificate Import Wizard window, click **Next** (Local Machine should be selected as the Store Location).
10. On the "**File to Import**" view, click the **Browse** button to locate and select the <server name>_RootCA.cer file. Click **Open**.

11. Click **Next**. On the “**Certificate Store**” view, the “**Place all certificates in the following store**” option should be selected, and “**Trusted Root Certificate Authorities**” should be automatically populated in the **Certificate store** field.
12. Click **Next**, then click **Finish**. A message should pop stating “**The import was successful.**”
13. To import the PKCS12 file, in the main MMC window, right-click on the **Personal** node under the **Certificates** node, mouse-over All Tasks, and select **Import**.
14. On the Certificate Import Wizard window, click **Next** (Local Machine should be selected as the Store Location).
15. On the “**File to Import**” view, click **Browse** to locate and select the <server name>_Cert.pfx file. Change the file type drop-down from the default “X.509 Certificate (*.cer;*.crt)” to “Personal Information Exchange (*.pfx;*.p12)”. Find and select the .pfx file, and click **Open**.
16. Click **Next**. On the “**Private key protection**” view, enter in the password specified in the pvk2pfx command-line options from step 1.5 in the previous section.
17. Click **Next**. On the “**Certificate Store**” view, the “**Place all certificates in the following store**” option should be selected, and “**Personal**” should be automatically populated in the Certificate store field.
18. Click **Next**, then click **Finish**. A message should pop stating “**The import was successful.**”
19. Close the MMC window.
20. Repeat these steps for every Call Recording Server.

Upload certificates to the Cisco PBX: On the PBX side, both the RootCA certificate and the service certificate need to be uploaded. If there is more than one Call Recording Server in the environment, these steps will need to be run to import certificates for all server. To upload the certificates, do the following:

1. Copy the <server name>_Cert.cer and <server name>_RootCA.cer to a location that is accessible from the Cisco PBX.
2. Log into the Cisco Unified Communications Operating System site with a user that has admin-level privileges.
3. For each of the two .cer files, do the following steps:
 - a. Navigate to **Security > Certificate Management**. The Certificate List window displays.
 - b. Click **Upload Certificate**. The Upload Certificate/Certificate Chain dialog box opens.
 - c. Select “**CallManager-trust**” from the Certificate Type list.
 - d. Click **Browse**, navigate to the certificate file, and click **Open**.
 - e. Click **Upload File**.
 - f. Repeat for every Call Recording Server’s certificates.

QMS High Availability and Forked Audio

QMS High Availability functionality allows for two Call Recording Services to be linked together, such that should the primary Call Recording Service become unavailable, the secondary Call Recording Service will take over the recording of phone calls. Because of the interactions between CUCM, the TSP, and the QMS server, some extra additions and configurations will be required on CUCM.

On the CUCM side, a unique SIP Trunk, SIP Trunk Security Profile, and Application User must be defined for each QMS server in the High Availability cluster. The Destination Ports of the SIP Trunks must also be unique for each QMS server. Even though each QMS server will have its own unique components defined in CUCM, CUCM will only use one SIP Trunk at a time to send the forked audio to the QMS cluster. This means that only one node in the QMS cluster will receive audio for a given call. The other piece in CUCM that needs to be configured is a Route List, which contains the two SIP Trunks. When a call comes in, CUCM will try to establish a connection with the TSP assigned to the first SIP Trunk in the Route List. If a connection can't be made, CUCM will try to make a connection with the second SIP Trunk in the Route List. Therefore, by configuring the primary Call Recording Service's TSP to connect to CUCM via the first SIP Trunk, and setting the secondary Call Recording Service's TSP to connect to CUCM via the second SIP Trunk, the High Availability setup is complete.

Due to the fact that CUCM will only use one SIP Trunk to send the forked audio to the QMS cluster, there is a limitation inherent to QMS High Availability in a Cisco Forked Audio environment. If a call is being recorded on the primary Call Recording Server, and the primary Call Recording Server goes down, that recording will be terminated at that point. CUCM will not know if a Call Recording Server goes offline, therefore it will not know to switch to the second SIP Trunk for sending the forked audio for that call. The next call CUCM handles will have its forked audio sent to the second SIP Trunk to be recorded by the secondary Call Recording Service, because CUCM will not be able to establish a connection to the TSP on the primary Call Recording Service.

PBX Configuration

1. Log into the CUCM Administration site as a user with admin-level privileges.
2. Follow step 3 under **PBX Configuration** in the Forked Audio – IP Phone-based Recording section above to create a SIP Trunk Security Profile for the primary SIP Trunk.
 - a. The default Incoming Port is **5060**. If this port is already being used, enter in a new value, otherwise accept the default value.
3. Follow step 3 under **PBX Configuration** in the Forked Audio – IP Phone-based Recording section above to create a SIP Trunk Security Profile for the secondary SIP Trunk.
 - a. The default Incoming Port is **5060**. The Incoming Port must be unique, so change the Incoming Port to something other than 5060. The default port for a secured SIP Trunk is 5061, so to avoid confusion, enter a port above **5061**.
 - b. Click **Save**.
4. Follow step 5 under **PBX Configuration** in the Forked Audio – IP Phone-based Recording section above to create a **primary** SIP Trunk.
 - a. Set the Destination Port in the SIP Information section to the **Incoming Port** value set in step 2 above.
5. Follow step 5 under **PBX Configuration** in the Forked Audio – IP Phone-based Recording section above to create a **secondary** SIP Trunk.
 - a. Set the Destination Port in the SIP Information section to the **Incoming Port** value set in step 3 above.
6. Create a Route Group.
 - a. Click the **Call Routing** menu, then click **Route/Hunt**, and finally click on **Route Group**.

- b. Click **Add New**.
 - c. Enter in a **Route Group Name**.
 - d. Select **Top Down** from the Distribution Algorithm drop-down list.
 - e. In the Available Devices list in the Route Group Member Information section, select the **primary** SIP Trunk created in step 4 above.
 - f. Click **Add to Route Group**.
 - g. In the Available Devices list in the Route Group Member Information section, select the **secondary** SIP Trunk created in step 5 above.
 - h. Click **Add to Route Group**.
 - i. Click **Save**.
7. Create a Route List.
 - a. Click the **Call Routing** menu, then click **Route/Hunt**, and finally click on **Route List**.
 - b. Click **Add New**.
 - c. Enter a Name and Description for the Route List.
 - d. Select the appropriate Cisco Unified Communications Manager Group from the drop-down list.
 - e. Check the **Enable this Route List** checkbox.
 - f. Click the **Add Route Group** button in the Route List Member Information section.
 - i. The Route List Detail Configuration view is displayed.
 - ii. Select the **Route Group** created in step 6 above from the Route Group drop-down list.
 - iii. Click **Save** to return to the Route List Configuration view.
 - g. Click **Save** on the Route List Configuration view.
 - h. Click **Apply Config**.
8. Create a Route Pattern
 - a. Click the **Call Routing** menu, then click **Route/Hunt**, and finally click on **Route Pattern**.
 - b. Click **Add New**.
 - c. Enter a unique **Route Pattern**.
 - d. Select a **Route Partition** and enter a **Description**, if necessary.
 - e. Select the **Route List** created in step 7 above in the Gateway/Route List drop-down.
 - f. For the **Route Option**, select the **Route this pattern** radio button.
 - g. Set all other values as necessary.
 - h. Click **Save**.
9. Follow step 4 under **PBX Configuration** in the Forked Audio – IP Phone-based Recording section above to create a Recording Profile.
 - a. Enter in the **Route Pattern** created in step 8 above as the **Destination Address**.
10. Follow step 6 under **PBX Configuration** in the Forked Audio – IP Phone-based Recording section above to configure IP phones to use the Recording Profile created in step 9 above.

QMS Configuration

1. Follow the steps outlined in the QMS High Availability guide to configure the High Availability cluster nodes.
2. Once the initial High Availability configuration is completed, the secondary Call Recording Server needs to have the following setting updated.
 - a. On the secondary Call Recording Server, open the CallRecordingService.exe.config file located in the QMS installation folder.
 - b. In the <Recording /> section, add key **ForkedAudioSIPPort**. The key name is case-sensitive.
 - c. Set the value of ForkedAudioSIPPort to the value of the **Incoming Port** specified for the **secondary** SIP Trunk Security Profile in step 3 under the PBX Configuration section above. An example config file could look like this:
`<Recording ServiceID=" 93490d53-83f4-45b3-b3bd-72a64fd6085d" LastConversionTime="01/08/2016 13:06:21" ForkedAudioSIPPort="5062" />`
3. If the default Incoming Port value for the primary SIP Trunk Security Profile was set to a value other than 5060 in step 2 under the PBX Configuration section, repeat step 2 in this section on the primary Call Recording Service

QMS Cisco Phone Controls

The QMS Cisco Phone Controls is an XML-based application, hosted as an application within Internet Information Services (IIS), that provides a menu of call recording control options for a Cisco IP Phone. A phone service is defined within Cisco UCM, which is then added to each phone's Subscribed Cisco IP Phone Services list as needed. The functionality provided by the Phone Controls will work if the QMS user has the necessary permissions to perform each action (start/stop/pause recording).

IIS Configuration

The first step in setting up the QMS Cisco Phone Controls is to create a Web Application in IIS to host the application.



When upgrading a previous installation of the Cisco Phone Controls, simply do steps 3-5 under Add an application to IIS below, copying the files into the existing folder under `inetpub\wwwroot` that contains the previous version. Restart the **World Wide Web Publishing Service** to reinitialize the app with the new files. All other steps below can be skipped.

1. Add an application to IIS:
 - a. On the server that will host the Phone Controls application, open a Windows Explorer window and navigate to the `inetpub\wwwroot` folder.
 - b. In the `inetpub\wwwroot` folder, create a folder named **QMSCiscoPhoneControls**.
 - c. Extract the contents of the QM Suite (or QM Suite with Prerequisites) zip file into a temporary folder.
 - d. Navigate down to the **Installers\Cisco Support\Cisco Phone Controls** folder.
 - e. Copy the contents of the Cisco Phone Controls folder into the **QMSCiscoPhoneControls** folder.
 - f. Start **IIS Manager**.
 - g. Navigate down to **Default Web Site** under the Connections pane.
 - h. Right-click on Default Web Site, and click **Add Application**.
 - i. In the Alias field, enter **QMSCiscoPhoneControls**.
 - j. Click **Select...** next to the Application pool field.
 - k. From the Application pool drop-down, select a pool that supports **.NET CLR Version 4.0** and has Pipeline mode of **Integrated**.
 - l. Click **OK**.
 - m. Click the **browse** button next to the Physical path field.
 - n. Navigate to the `inetpub\wwwroot` folder.
 - o. Select the **QMSCiscoPhoneControls** folder.
 - p. Click **OK**.
2. Verify the application was created successfully:
 - a. Under the Default Web Site in IIS Manager, right-click **QMSCiscoPhoneControls**.
 - b. Click **Manage Application**.
 - c. Click **Browse**.

- d. A web browser window should open to URL **http://localhost/QMSCiscoPhoneControls**.
- e. The web page should display **QMS Controls for Cisco IP Phones**.
3. Update Security settings for logging:
 - a. Open a Windows Explorer window, and navigate down to the `inetpub\wwwroot` folder.
 - b. Right-click the `QMSCiscoPhoneControls` folder and click **Properties**.
 - c. Click the **Security** tab.
 - d. Click **Edit...**
 - e. If the **IIS_IUSRS** user is already in the list under Group or user names click on it to select it, and skip to Step j.
 - f. Click **Add**.
 - g. In the Enter the object names to select field, type **IIS_IUSRS**.
 - h. Click **Check Names**, which should cause **IIS_IUSRS** to be underlined.
 - i. Click **OK**.
 - j. Check the **Allow** checkbox for the Modify permission.
 - k. Click **OK** on the Permissions for `QMSCiscoPhoneControls` window.
 - l. Click **OK** on the `QMSCiscoPhoneControls` Properties window.
4. Update the `Web.config` settings:
 - a. Open a Windows Explorer window, and navigate down to the `inetpub\wwwroot\QMSCiscoPhoneControls` folder.
 - b. Open the **Web.config** file in a text editor.
 - c. Under `<appSettings>`, find the **logLevel** key.
 - d. Set the **logLevel** key to the appropriate level. Normal production systems should have this set to 3 or 4. When troubleshooting, this value should be set to 0, 1, or 2, depending on how much information is to be collected.
 - e. The `loginKey`, `passwordKey`, and `extensionKey` values are the corresponding Parameter Names for the Cisco IP Phone Service as described below. The default values should be sufficient, unless there is a specific need to change them.
 - f. The username and password values should be set to the credentials of a QMS user that has permission to start/stop/pause recordings.
 - g. Save any changes and close the `Web.config` file.
 - h. Restart the **World Wide Web Publishing Service** in Windows Control Panel -> Administrative Tools -> Services to pick up any changes made to the `Web.config` file, and to make sure the Phone Controls application is initialized properly.

PBX Configuration

Configuration in Cisco UCM for the QMS Cisco Phone Controls occurs in two main areas: IP Phone Services, and Phone Devices. The IP Phone Services configuration must be completed first, followed by configuring the IP Phone devices that need the Phone Controls.

1. IP Phone Services Configuration:
 - a. Log into the Cisco UCM Administration site as a user with admin-level privileges.
 - b. Click on the **Device** menu.

- c. Click **Device Settings**.
 - d. Click **Phone Services**.
 - e. Click the **Add New** button to bring up the IP Phone Services Configuration view.
 - f. In the **Service Name** field, enter QMS Phone Controls.
 - g. In the **ASCII Service Name** field, enter QMS Phone Controls.
 - h. In the **Service Description** field, enter QMS Phone Controls for Cisco IP Phones.
 - i. In the **Service URL** field, enter
`http://<hostname_or_IP_address>/QMSCiscoPhoneControls/main.aspx`.
 - j. In the **Service Category** drop-down, select XML Service.
 - k. In the **Service Type** drop-down, select Standard IP Phone Service.
 - l. Check the **Enable** checkbox.
 - m. In the **Service Parameter Information** area, three parameters need to be added:
 - i. Click **New Parameter**.
 - ii. In the **Parameter Name** field, enter **ext**.
 - iii. In the **Parameter Display Name** field, enter **Agent Extension**.
 - iv. In the **Parameter Description** field, enter **QMS Agent Extension**.
 - v. Check the **Parameter is Required** checkbox.
 - vi. Click **Save and Close**.
 - vii. Click **New Parameter**.
 - viii. In the **Parameter Name** field, enter **login**.
 - ix. In the **Parameter Display Name** field, enter **Login Name**.
 - x. In the **Parameter Description** field, enter **Optional Login Name for Service Authentication**.
 - xi. Click **Save and Close**.
 - xii. Click **New Parameter**.
 - xiii. In the **Parameter Name** field, enter **password**.
 - xiv. In the **Parameter Display Name** field, enter **Password**.
 - xv. In the **Parameter Description** field, enter **Optional Password for Service Authentication**.
 - xvi. Check the **Parameter is a Password** checkbox.
 - xvii. Click **Save and Close**.
 - n. Click **Save** on the IP Phone Services Configuration view.
 - o. Click **Update Subscriptions**.
2. IP Phone Devices Configuration:
- a. Log into the Cisco UCM Administration site as a user with admin-level privileges.
 - b. Click on the **Device** menu.
 - c. Click **Phone**.
 - d. In the **Find and List Phones** view, enter search criteria if needed, and click **Find**.
 - e. Find the phone that needs the Phone Controls from the list, and click on the corresponding link in the **Device Name** column.

- f. In the **Phone Configuration** view, select **Subscribe/Unsubscribe Services** in the **Related Links** drop-down.
 - g. Click **Go**, which will cause a **Subscribed Cisco IP Phone Services** pop-up window to appear for the phone.
 - h. In the **Select a Service** drop-down, select **QMS Phone Controls**.
 - i. Click **Next**.
 - j. Enter the appropriate value in the **Agent Extension** field, which is required.
 - k. Enter values in the **Login Name** and **Password** fields, if required.
 - l. Click **Subscribe**. QMS Phone Controls should appear in the **Subscribed Services** section.
 - m. Click **Save**.
 - n. Close the **Subscribed Cisco IP Phone Services** window.
3. Verify the Phone Controls are accessible on the IP Phone's Services menu:
 - a. Press the **Services** button on the phone.
One of the Services should be **QMS Phone Controls**.

Assigning Cisco Phone Controls to an IP Phone's Hard Key (optional)

From a usability perspective, it can be advantageous to assign the Cisco Phone Controls to a Cisco IP Phone's hard key, if one is available. Assigning the Cisco Phone Controls to a hard key (or speed dial button, in this case) is also the only way an IP Phone that does not have a Services button can use the Controls. This section outlines the steps needed to add the Cisco Phone Controls to an available hard key or speed dial button on a Cisco IP Phone.

1. IP Phone Devices Configuration:
 - a. Log into the Cisco UCM Administration site as a user with admin-level privileges.
 - b. Click on the **Device** menu.
 - c. Click **Phone**.
 - d. In the **Find and List Phones** view, enter search criteria if needed, and click **Find**.
 - e. Find the phone that needs the Phone Controls from the list, and click on the corresponding link in the **Device Name** column.
 - f. Under the **Association Information** section, click the **Add a new SURL** link.
 - g. From the **Button Service** drop-down, select **QMS Phone Controls**.
 - h. The **Label** and **ASCII Label** fields auto-populate with QMS Phone Controls. It is recommended to shorten the two labels to **QMS Controls**.
 - i. Click **Save**, then click **Close**.
 - j. Back on the Phone Configuration page, click **Modify Button Items**.
 - k. In the **Associated Items** box, select an item from the list that is to be replaced. Make sure no "Line" entries are selected, unless a secondary line is not needed.
 - l. Click the **Down Arrow** under **Associated Items**, to move the selected item to the **Dissociate These Items** box.
 - m. In the **Unassigned Associated Items** box, select **QMS Controls**.
 - n. Click the **Left Arrow** between the Associated Items and Unassigned Associated Items boxes to move QMS Controls over to the Associated Items box.

- o. The **Up** and **Down Arrows** between the Associated Items and Unassigned Associated Items boxes can be used to position QMS Controls in the desired hard key on the IP Phone.
- p. Click **Save**, then click **Close**.
- q. Verify the corresponding hard key on the IP Phone displays **QMS Controls**.

CUCM User Options Configuration (optional)

This section describes the rare case of agents configuring their own devices. Such agents will have their own CUCM user account, and are responsible for administering their own phone devices. The following directions can be distributed to the agents that fall under this category.



Depending on the version of Cisco UCM installed, the steps needed to complete the configuration may be slightly different from those specified below.

1. Configuring device options:

- a. Log into the Cisco UCM User Options site with your UCM user credentials.
- b. Click **Phone Services**. If Phone Services does not appear on the toolbar, add the **Standard CCM Phone Administrator** group to your UCM user account.
- c. Click **Add New** on the Find and List IP Phone Services page.
- d. In the **Select a Service** drop-down, select **QMS Phone Controls**.
- e. Click **Next**.
The IP Phone Service Configuration page will display, containing the Agent Extension, Login Name, and Password fields.
- f. Enter your extension in the **Agent Extension** field.
- g. If you have credentials for the QMS Client, enter your QMS username and password in the **Login Name** and **Password** fields, respectively. These two fields are optional. If they are left blank, the user credentials defined in the QMS Cisco Phone Controls config settings will be used.
- h. Click **Save**

QMS Configuration File Settings (Optional)

The following are Cisco-specific settings that can be added if necessary to the CallRecordingService.exe.config file, located in the QMS installation folder. All key names are case-sensitive.

DisableDHCP: Disables processing of IP address updates in a Forked Audio recording environment for QMS Users. Valid values are true and false, with a default value of false. This key is located in the **<System />** section.

EMLine: For Cisco IP phones running SCCP, the Extension Mobility extension/line to compare current line number against to see if an IP address update has occurred. Valid values are any valid integer, with a default value of 1. This key is located in the **<System />** section.

SipServerTlsCertificateFilePath: Indicates the file path/name of the certificate file that will enable QMS to act as a secure SIP server. Used with secure forked audio recording. Valid values include any string that indicates a directory path and file name to an existing certificate file, with no default value. This key is located in the **<System />** section.

CiscoFADynamicAudioPortBase: The first audio port, in a range of ports, to be used in Forked Audio – Network-based recording dynamic port assignment. Valid values are the range of integers from 1 – 65535, although ports at or below 1024 are generally reserved for system processes. The default value is 16000. This key is located in the **<Recording />** section.

CiscoFADynamicAudioPortRange: The number of ports available for Forked Audio – Network-based recording dynamic port assignment. This value should be the number of QMS users * 10. Valid values are the range of integers that, when added to CiscoFADynamicAudioPortBase, are between 1 – 65535. The default value is 2000. This key is located in the **<Recording />** section.

DisableCiscoFADynamicAudioPorts: Indicates whether QMS will employ a unique audio port to receive each stream in a Forked Audio – Network-based recording environment. A unique port is necessary when all audio is coming from a single source IP, for example from a single Cisco Voice Gateway, in order to associate the correct QMS user to a given audio stream. Valid values are true and false, with a default value of false. This key is located in the **<Recording />** section.

DisableCiscoFADynamicAudioPortUserPersistence: In a Forked Audio – Network-based recording environment, which uses unique audio ports to associate QMS users to audio streams, the default behavior is to maintain port-user associations between calls. Setting this value to true disables that behavior, so the ports associated to a given QMS user are returned to available ports pool once the user state returns to Idle. Valid values are true and false, with no default value. This key is located in the **<Recording />** section.

ForkedAudioAdapterId: Specifies the network adapter to be used as the forked audio endpoint on the QMS server, which will receive SIP messages and RTP audio from the PBX. Valid values include any valid GUID, with an empty GUID as the default value. This key is located in the **<Recording />** section.

ForkedAudioRTPPort: Specifies the port to use for receiving RTP audio at the QMS server, in a Forked Audio – IP Phone-based recording environment. Valid values are the range of integers from 1 – 65535, although ports at or below 1024 are generally reserved for system processes. The default value is 8900. This key is located in the **<Recording />** section.

ForkedAudioSecureRTPPort: Specifies the port to use for receiving secure RTP audio at the QMS server, in a Forked Audio – IP Phone-based recording environment. Valid values are the range of integers from 1 – 65535, although ports at or below 1024 are generally reserved for system processes. The default value is 8904. This key is located in the **<Recording />** section.

ForkedAudioSIPPort: Specifies the port to use for receiving SIP messages at the QMS server, in a Forked Audio – IP Phone-based recording environment. Valid values are the range of integers from 1 – 65535, although ports at or below 1024 are generally reserved for system processes. The default value is 5060. This key is located in the **<Recording />** section.

ForkedAudioSIPPortSecure: Specifies the port to use for receiving secure SIP messages at the QMS server, in a Forked Audio – IP Phone-based recording environment. Valid values are the range of integers from 1 – 65535, although ports at or below 1024 are generally reserved for system processes. The default value is 5061. This key is located in the <Recording /> section.

LogForkedAudioSIPMsgs: Indicates whether SIP messages should be logged in either type of Forked Audio recording environment. Valid values are true and false, with a default value of false.

DisableCiscoDHCP: Disables processing of IP address updates from SCCP messages for QMS Users. Valid values are true and false, with a default value of false. This key is located in the <Control /> section.

DisableSkinnyTracing: Turns on/off logging of SCCP (Skinny) messaging. Valid values are true and false, with a default value of false. This key is located in the <Control /> section.

EnableCiscoTapiKeyDelivery: Indicates whether Cisco DevSpecific information will be utilized for SRTP key delivery. Valid values are true and false, with a default value of false. This key is located in the <Control /> section.

EnableVerboseSkinnyTracing: Turns on/off logging of all SCCP (Skinny) messages, except KeepAlive messages (the default SCCP tracing is only log-processed messages). Valid values are true and false, with a default value of false. This key is located in the <Control /> section.

RecordBusySignals: Specifies whether recording of calls that result in busy signals are to be recorded. Valid values are true and false, with a default value of false. This key is located in the <Control /> section.

UseLastRedirectingPartyNumberForDialedDigits: In an SCCP (Skinny) environment, this key indicates that the LastRedirectingPartyNumber value is to be used for dialed digits, rather than the default of CalledPartyNumber. Valid values are true and false, with a default value of false. This key is located in the <Control /> section.

UseOriginalCalledPartyNumberForDialedDigits: In an SCCP (Skinny) environment, this key indicates that the OriginalCalledPartyNumber value is to be used for dialed digits, rather than the default of CalledPartyNumber. Valid values are true and false, with a default value of false. This key is located in the <Control /> section.

Troubleshooting

As initial installation finishes, Cisco TSP installer shows error “Could not load the dll file EncryptPassword.dll”.

1. Re-run installer.
2. Choose **Reinstall**.
3. Click **Next**, then reboot the QMS server when prompted.

After successful installation, the TSP Notifier icon is not present in the Notifications area.

1. Reboot the QMS server.

No TSP profile listed in the TSP Notifier app, or the TSP Notifier app crashes after right-clicking its icon and selecting Cisco TSP Configuration.

1. Start Control Panel.
2. Double-click **Phone and Modem**.
3. Click the **Advanced** tab.
4. Click **Add** button.
5. Select **CiscoTSP001.tsp**.
6. Click **Add** button.
7. Select **CiscoTSP001.tsp** in the Providers list
8. Click **Configure** button.
9. Enter correct info in the **User**, **CTI Manager**, and **Security** tabs where needed
10. Click **OK**, then close Phone and Modem app.
11. Verify changes were saved by right-clicking the **Cisco TSP Notifier** app and clicking **Cisco TSP Configuration** on the menu. CiscoTSP001.tsp should appear on the list. Highlight it and click the Configure button.
12. Verify information in the User, CTI Manager, and Security tabs was saved correctly.

Phones in encrypted media environment are not registering as secure devices

Verify CUCM has been set up to allow devices to register as secure devices:

1. Log into CUCM Administration site.
2. Click on the **System** menu
3. Click on **Enterprise Parameters**.
4. Scroll down to the **Secure Parameters** section
5. Check the value of the **Cluster Security Mode**.

If the value is 0, CUCM has not been configured to allow secure device registration. Follow the steps outlined in the Media Encryption section, PBX Configuration, section 1.

After logging into the QMS server, the TSP initialization fails with a “bind socket failed” and/or an “Initializing Socket to listen on port xxxx failed” error.

This problem typically only occurs when multiple users are logged onto the QMS server simultaneously, either locally or through a Remote Desktop session, causing the TSP to initialize multiple times. Make sure only one user logs onto the QMS server at a time.

A QMS User configured with multiple extensions in a Forked Audio environment is not recording.

Verify the QMS User's phone is configured properly in CUCM. See #6 under **PBX Configuration** in the Forked Audio – IP Phone-based Recording section.

A QMS User configured with multiple extensions in a Forked Audio environment is not recording, but the user's IP phone has the Built-In-Bridge enabled.

Verify the QMS User's list of extensions.

1. The QMS User's primary extension must be the line 1 appearance in CUCM.
2. The QMS User's primary extension cannot be the line 1 appearance for any other user in CUCM.
3. Any other line appearances configured for the QMS User in CUCM must be listed in the Additional Extensions field, separated by commas, in the Users view of the QMS Client for the QMS User.
4. The complete list of extensions for a QMS User, referred to as the User's phone signature, must be unique. Another QMS User cannot have the same list of extensions, even if the extensions are listed in a different order. For example:
 - a. User 1 has a primary extension of 101, and Additional Extensions of 102 and 103. User 1's phone signature is [101 102 103].
 - b. User 2 has a primary extension of 102, and Additional Extensions of 103 and 101. User 2's phone signature is [102 103 101].
 - c. The phone signatures are duplicates, because they both contain the same list of extensions.
 - d. To resolve, update CUCM configurations to make sure all QMS Users have a unique extension assigned that is not shared.

The QMS Client Real-Time Activity view is showing users in an Idle state when calls are actively going on, in a Forked Audio configuration with QMS High Availability.

In situations where the primary Call Recording Service goes down, user states are reset to Idle after the Call Recording Service comes back up. A user state change must come from CUCM after the Call Recording Service restarts for Real-Time Activity to reflect the correct current state. This is not a bug, it is a limitation of the configuration necessary in CUCM to allow Forked Audio to work with a QMS High Availability cluster.

In a Forked Audio environment with “Create Dual Channel Recordings” enabled on the Call Recording Service, all of the audio is on only one channel.

This is not a bug, but a limitation in the way Forked Audio sends audio packets to the QMS server. For stereo encoding to create a dual-channel recording, two source IP addresses are needed, in order to determine which audio packets should go into each respective recording channel. Forked Audio will send audio packets to the QMS server either from an IP phone, or from a network gateway device. In either case, all audio is coming from one source IP address, which is associated with the recorded user. The result is all audio ends up on a single channel.

QMS does not detect or does not record calls.

The SIP Profile, for either a Non-HA or HA environment, must have SIP OPTIONS Ping disabled for each configured trunk. This setting has been known to interfere with SIP call control which can result in calls not being detected or not recorded.

Terms of use

Any software ("Software") that is made available by Enghouse Interactive Inc. ("Enghouse"), together with any User Documentation ("User Documentation") is the copyrighted work of Enghouse. Use of the Software is governed by the terms of a Master Purchase Agreement, End User License Agreement, or similar software license agreement ("License Agreement"). End users are not legally authorized to install any Software that is accompanied by or includes a License Agreement unless he or she first agrees to the License Agreement terms.

The Software is made available for installation solely for use by users according to the License Agreement. Any reproduction or redistribution of the Software not in accordance with the License Agreement is expressly prohibited by law and may result in severe civil and criminal penalties. Violators will be prosecuted to the maximum extent possible.

WITHOUT LIMITING THE FOREGOING, COPYING OR REPRODUCTION OF THE SOFTWARE TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION IS EXPRESSLY PROHIBITED, UNLESS SUCH REPRODUCTION OR REDISTRIBUTION IS EXPRESSLY PERMITTED BY THE LICENSE AGREEMENT ACCOMPANYING SUCH SOFTWARE.

THE SOFTWARE IS WARRANTED, IF AT ALL, ONLY ACCORDING TO THE TERMS OF THE LICENSE AGREEMENT. ENGHOUSE HEREBY DISCLAIMS ALL OTHER NON-EXPRESS WARRANTIES AND CONDITIONS WITH REGARD TO THE SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT.

Enghouse grants a nonexclusive license to customer for use of the User Documentation. The User Documentation contains copyrighted and other proprietary materials. By accepting the User Documentation, recipients agree that they will not transmit, reproduce, or make available to any external third-party this User Documentation or any information contained herein. Copying, reverse-engineering, or reselling any part of the Software or User Documentation is strictly prohibited.

The information contained in the User Documentation furnished by Enghouse is based on the most accurate information available at the time of printing. No representation or warranty is made by Enghouse as to the accuracy or completeness of such information or any ongoing obligation to update such information. Enghouse reserves the right to change the information contained in this document without notice.

Registered Trademarks

Syntellect®, Voyager®, Continuum®, MediaVoice®, Apropos®, Envoy®, Envoy® Activecall, Envoy CT ADE®, Envoy CT Connect®, Dynamic Application Discovery®, Interaction Vault® CT Impact®, SmartDialer®, SmartVoice®, SmartCollect®, SmartSupport®, Zeacom®, Enghouse Systems®

End User License Agreement

1. **Applicable Law; Definitions.** This End User License Agreement ("Agreement"), the definition of terms used, performance hereunder, and the interpretation of this Agreement shall be governed by and construed in accordance with the laws, other than the conflicts of laws rules, of the state of Arizona. If the Uniform Computer Information Transaction Act ("UCITA") is enacted as part of the law of the state of Arizona, such statute will not govern any aspect of this Agreement, any license granted hereunder or any of the parties' rights and obligations arising under this Agreement. "Confidential Information" means any non-public information or documentation provided by Supplier under this Agreement, including but not limited to Software, documentation, and information pertaining thereto. "End- User Customer" means the final licensor of Software who licenses for their use. "Agreement" means this document. "Software" means Supplier's proprietary Software and any third party proprietary software components licensed to Customer pursuant to this Agreement. "Terms" means, collectively, the terms and conditions set forth below and on the front of this Agreement.
2. **Agreement.** This Agreement constitutes an acceptance of Customer's offer to license Software contingent and consistent upon the Terms contained herein. Any terms or conditions proposed by Customer inconsistent with or in addition to the Terms shall be void and of no effect, unless specifically agreed to in a signed writing by an authorized Supplier representative. Payment for Software by Customer or furnishing of the Software by Supplier, in whole or in part, shall constitute a binding agreement on the Terms of this Agreement. The failure of Supplier to insist upon performance of this Agreement, to enforce any of the Terms of this Agreement or other purchase orders from Supplier, or to exercise any right or privilege granted to Supplier under this Agreement or under law, shall not be construed as a waiver and the same shall continue in full force and effect.
3. **License Grant.** Subject to Customer's compliance with the terms of this Agreement, Supplier grants to Customer a non-exclusive, non-transferable, perpetual license to install, use and execute the Software in object code form on a per-license basis consistent with the licensing scheme applicable to the type of software purchased on this Agreement at the location specified on the PSOF ("Software License") as may be changed by Customer from time to time upon prior written notice to Supplier, such Software License limited to the site(s), number of seats, concurrent users, agents, servers, ports, devices, managed applications, and/or copies as applicable to the Software obtained, not to exceed the number of licenses set forth on PSOFs placed pursuant to this Agreement. The Software License shall become effective upon Delivery of the Software and shall remain in force unless terminated pursuant to Section 12 of this Agreement. This right does not include permission to grant sub-licenses or otherwise transfer such rights. The Customer may make one (1) copy of the Software for non-productive archival purposes only, provided that it retains or affixes the equivalent of Supplier's proprietary legend and copyrights to the copy. Additionally, the Customer may make several copies of the system documentation, excluding training manuals and materials, provided that they are for internal use only. Customer may not reverse engineer, disassemble or otherwise translate the Software provided pursuant to this Agreement. Supplier, or any third party that owns the Software License, retains exclusive title to and all rights to the Software. The Customer acknowledges that the Software and documentation are the property of Supplier and that the only right that the Customer obtains to the Software is the right of use in accordance with the terms of this Agreement. To assist Supplier in the performance of its duties under this Agreement and in the protection of its proprietary rights, Customer hereby authorizes a Supplier representative to enter Customer's premises, physically or electronically, and inspect the Software License at reasonable times with prior notice.
4. **Compliance with Laws.** Customer shall comply with all applicable laws, including, without limitation, the export control laws and regulations of the United States of America and those countries involved in transactions concerning the exporting of Software purchased under the terms and conditions or the control or regulation of the exportation of products or technical data supplied to Customer by Supplier. Customer will be responsible to pay all taxes, tariffs and duties. Customer shall comply with the United States Foreign Corrupt Practices Act and shall indemnify Supplier from Customer's violations of

such Act. The United Nations Convention on the International Sale of Goods (CISG) is specifically excluded and shall not apply to any transaction contemplated herein.

5. **Limited Warranty.** Supplier warrants to Customer that Software will perform in every material respect according to the published specifications for a period of thirty (30) days from Delivery. However, neither Supplier nor its third-party supplier's warrant Software will meet Customer's requirements or that the Software operation will be uninterrupted or error-free. Remedies for Software defects reported during the warranty period consist of (and are limited to), at Supplier's option, repairing, replacing or refunding the purchase price of the Software. This warranty is contingent upon the proper use and application of the Software in accordance with Supplier's instructions. The warranty does not (i) cover the Software if modified by anyone other than Supplier; (ii) apply if Software warranty support is requested as a result of accident, neglect, or operating conditions exceeding specifications; (iii) cover malfunctions caused by defects in or incompatibility to Customer's computer system or equipment; (iv) cover malfunctions caused by defects in or arising from the installation, repair, or programming of the Software other than by Supplier; and (v) apply if Customer has rejected or not used any Software corrections, updates, or modifications supplied or made available by Supplier. THE WARRANTIES HEREIN DO NOT APPLY TO THIRD-PARTY SOFTWARE FURNISHED BY SUPPLIER UNDER THIS AGREEMENT, AND SUCH PRODUCTS ARE PROVIDED ON AN "AS IS" BASIS. AS APPLICABLE, SUPPLIER AGREES TO ASSIGN ANY WARRANTY IT MAY HAVE WITH RESPECT TO THIRD-PARTY SOFTWARE TO CUSTOMER, AND CUSTOMER AGREES TO PROCEED DIRECTLY AND EXCLUSIVELY AGAINST THE THIRD-PARTY SUPPLIER AS TO ANY CLAIMS OF WARRANTY. THE FOREGOING WARRANTIES ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
6. **Limitation of Liability.** SUPPLIER'S LIABILITY FOR DAMAGES FOR ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF ACTION, SHALL BE LIMITED TO, AT SUPPLIER'S OPTION, REPAIR OR REPLACEMENT OF THE DEFECTIVE SOFTWARE. IN NO EVENT WILL SUPPLIER BE LIABLE FOR DAMAGES IN EXCESS OF the fees paid PURSUANT TO AN ORDER THAT FORMS THE BASIS OF THE CLAIM. IN NO EVENT SHALL SUPPLIER BE LIABLE OR RESPONSIBLE FOR ANY REPROCUREMENT COSTS, LOSS OF PROFITS, LOSS OF USE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES OF ANY KIND, WHETHER IN AN ACTION OF CONTRACT OR TORT, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. THIS LIMITATION OF LIABILITY AND EXCLUSION OF CERTAIN DAMAGES SHALL APPLY REGARDLESS OF THE SUCCESS OR EFFECTIVENESS OF OTHER REMEDIES. NO ACTION ARISING OUT OF OR IN CONNECTION WITH THIS AGREEMENT OR ANY TRANSACTION HEREUNDER MAY BE BROUGHT AGAINST SUPPLIER MORE THAN TWELVE (12) MONTHS AFTER THE CAUSE OF ACTION HAS ARISEN.
7. **Relationship of the Parties.** Each party acknowledges that they are independent contractors, and that the relationship between Supplier and Customer is that of manufacturer/Customer. Neither party shall in any way represent or obligate the other party to any contract, term, or condition, nor may it represent the other party as agent, employee, franchisee, or in any other capacity. Customer is expressly prohibited from making representations to other third parties regarding Supplier or the Software and Services in excess of or contrary to representations contained in Supplier's or its third-party supplier's product specifications, brochures, newsletters, and other publications provided to Customer by Supplier. Nothing in this Agreement shall be construed to limit either party's right to independently develop or distribute products which are functionally similar to the other party's products, so long as Confidential Information of the other party is not used in such development or distribution.
8. **Assignment.** Neither this Agreement nor any rights under it may be assigned by Customer without Supplier's prior written consent. Any unauthorized assignment by Customer shall be void. Supplier may assign the Agreement, in whole or in part, upon thirty (30) days' notice to the Customer.

9. **Severability.** If a court of competent jurisdiction holds any provision in this Agreement to be invalid, void, or unenforceable, the remaining provisions will yet continue in full force without being impaired or invalidated in any way
10. **Notices.** Any notices given to Supplier may be made either by facsimile, overnight courier, hand delivery, or by registered or certified mail, postage prepaid with return receipt requested. Mailed notices shall be addressed to Supplier at the address appearing below, but Supplier may change such address by written notice in accordance with this paragraph. Notices mailed, or delivered personally or by electronic means will be deemed communicated as of actual receipt or the date of transmission, as applicable.

Enghouse Interactive
216 Route 17 North, Suite 301
Rochelle Park, NJ 07662, United States

11. **Confidentiality.** The parties agree to accept in confidence all Confidential Information provided to them. The parties further agree not to use or disclose any Confidential Information supplied by the other except as required to perform their obligations in accordance with this Agreement. Any disclosure of Confidential Information to agents or employees shall be made only in the normal course of business, on a need-to-know basis, within the scope and purpose of this Agreement, and under written agreements requiring such agents or employees to treat all such information as strictly confidential. The parties agree not to use, publish, reproduce, disseminate, or otherwise disclose the other's proprietary or Confidential Information, including, but not limited to customer lists, without the prior written consent of the other.

Customer shall not develop, manufacture, maintain, or, except as contemplated under this Agreement, market products or services incorporating Supplier's Confidential Information or Software. Customer agrees to include Supplier's proprietary notice on all copies of Supplier's Confidential Information made by Customer and to maintain records of the location of these copies.

Customer agrees that the Software, and any proprietary and/or trade secret information and data furnished to Customer by Supplier or any of its third-party suppliers will be considered Confidential Information and subject to the provisions of this Section. Supplier reserves for itself all proprietary rights in all designs, engineering details, and other data pertaining to the Software and other proprietary data arising out of work done in connection with designing, manufacturing, servicing, and testing the Software, including the sole right to manufacture and, except as provided herein, market all such items. Customer shall not reverse, translate, disassemble, or decompile the Software or any component of the Software.

12. **Entire Agreement.** This Agreement is the entire agreement of the parties regarding the subject matter hereof, and supersedes and terminates any prior agreements, understandings or representations, written or oral, except with respect to any trade indebtedness owing between the parties.

Enghouse Systems welcomes user comments and reserves the right to revise this document and/or make updates to product specifications, products, or programs described without notice at any time. Enghouse Systems makes no representations or warranties regarding this document. The names of actual companies and products mentioned herein are the trademarks of their respective owners. Enghouse Interactive and all Enghouse Systems logos are trademarks or registered trademarks of Enghouse Systems and may be used publicly only with the written permission by an authorized company agent. Other listed names and brands are trademarks or registered trademarks of their respective owners.

© Copyright 1998-2021 by Enghouse Systems, Limited. All rights reserved.

No part of this publication may be reproduced without the prior written consent of Enghouse Systems.



Enghouse Interactive