Installation and Configuration Guide

AudioCodes WebRTC Solutions for Enterprises

WebRTC Click-to-Call Widget

Version 1.1

audiocodes

Table of Contents

Notice

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document can be downloaded from https://www.audiocodes.com/library/technical-documents.

This document is subject to change without notice.

Date Published: July-04-2022

Customer Support

Customer technical support and services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our Web site https://www.audiocodes.com/services-support/maintenance-and-support.

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

Related Documentation

Document Name
https://www.audiocodes.com/solutions-products/solutions/enterprise-voice/webrtc-connectivity

Document Revision Record

LTRT	Description
14030	Initial document release for Version 1.0
14031	Updated installation procedure
14032	Added advanced configuration and customization.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our Web site at https://online.audiocodes.com/documentation-feedback.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our website at https://online.audiocodes.com/documentation-feedback.

1 Introduction

WebRTC technology enriches user experience by adding voice, video and data communication to the browser, as well as to mobile applications. AudioCodes WebRTC gateway provides seamless connectivity between WebRTC clients and existing VoIP deployments.

A typical WebRTC solution comprises a WebRTC gateway, which is an integrated functionality on AudioCodes' SBCs, and a client application running on a browser or a mobile app.

AudioCodes' WebRTC Widget can easily be integrated into websites and blogs with basic JavaScript knowhow. The Widget creates a click-to-call button on your web site. It works with AudioCodes' SBC and can make calls to any user registered in the SBC.

1.1 WebRTC Benefits

Here's a summary of the WebRTC benefits:

- Simple deployment - a single WebRTC gateway device for both signaling and media.
- Strong security and interoperability capabilities resulting from integration with SBC.
- Client SDK for browsers.
- OPUS powered IP phones for superb, transcoder-less voice quality.
- Optional recording of WebRTC calls.

1.2 Click-to-call Phone (Widget)

The 'click-to-call' phone or widget (for outgoing calls only) uses the anonymous user mode on the AudioCodes SBC. The phone doesn't ask the user for any information and does not save anything in the browser. In doing so, it can be safely used on public computers, such as the ones available at airports, Internet cafes, or public libraries.

The phone call is initiated from an anonymous user to a user that is registered in the SBC. To use this phone, the webmaster inserts the widget on an HTML page. The phone is a single HTML page application. If you leave the page, the phone program is terminated.

The phone can only be used on an HTTPS site (WebRTC API security limitation).

2 Widget Installation

The procedure below describes how to install the widget on the HTTPS web server.

To install the widget on the HTTPS web server:

1. Unzip the click-to-call phone zip file; the following directories are extracted:
   - 'conf' (containing the configuration file)
   - 'css' (containing the demo style file)
   - 'js' (containing the SDK JS, audio player and widget script code)
   - 'html' (containing the phone HTML page example and icon pictures)
   - ‘sounds’ (containing sounds)
   - 'docs' (the phone documents)
2. Copy the directory’s tree (except for the ‘docs’ directory) to your HTTPS web server.
3. For the initial check of the phone, only change the configuration file. Edit the provided config.js file.

> Please edit only the **red** marked values keeping the file structure as is.

b. The first part (server config) contains Internet addresses:

- ♦ domain: AudioCodes SBC domain name (used to build SIP From/To headers)
- ♦ addresses: AudioCodes SBC secure web socket URL

```
let c2c_serverConfig = {
  domain: 'example.com',
  addresses: ['wss://sbc.example.com'],
  iceServers: []
};
```

c. The second part (phone config) contains the call and type:

- ♦ call: Call destination (User or phone number).
- ♦ type: Call type: ‘audio’, ‘video’ or ‘user_control’.

```
let c2c_config = {
  call: 'JohnDoe',
  …
  type: 'user_control',

  …
}
```

4. Check that the phone is working; confirm that you properly configure the microphone and speaker for using the browser and operating system (Windows, macOS, Linux, Android, iOS).
5. Open **youtube.com** in your browser, and then play a video to confirm that you hear sound.
6. Open the operating system microphone settings, and then confirm that the microphone works.
7. Open the phone HTML page using the Chrome browser: **https://<your site domain>/ c2c/html/**.

8. Open the browser console log (for Chrome - **Ctrl/Shift/I**, and then select the **Console** tab)
9. Press the phone **Call** button; the Console log shows the phone connecting to the SBC using a web socket and starts calling (i.e., it sends an initial SIP INVITE).
10. Answer the call and test the sound quality.

3 Phone URL Parameters

The DTMF sequence can only be set in the URL and not in the config.js file, and is sent after

after the call is opened.

This can be useful if the phone number used a menu selected by DTMF, or

some internal numbers for workers.

The following parameters can be taken from the URL, if the config.js file is assigned the “take_value_from_url” value:

- ’domain’
- ’server’
- ’call’
- ’logger’

For example:

https://example.com/c2c/html/?call=user17&delay=1000&dtmf=0123456789%23

- “call=" defines the callee user.
- "delay=" defines the delay (in milliseconds) before sending the DTMF.
- "dtmf=" defines the DTMF sequence.

4 Phone Configuration

Configure your phone settings as shown below.

```
let c2c_serverConfig = {
    domain: 'example.com',
    addresses: ['wss://sbc.example.com:443'],
    iceServers: [],
    // Optional websocket logger server (instead console.log)
    //logger: 'example.com/wslog'
};
```

- **domain** (string) : Defines the SBC domain name (used to build SIP From/To headers).

  Use the '_take_value_from_url_' value to take this value from the URL ‘domain’ parameter.
- **addresses** (string array or string '_take_value_from_url_')

  Defines the SBC WebSocket URL(s). Use the '_take_value_from_url_' value to take this value from the URL ‘server’ parameter.
- (Optional) **iceServers** (String array) – Defines the STUN and TURN server URL(s). When using the SBC, the servers are not needed to bypass NAT. An empty array is used.
- (Optional) **logger** (String) Defines the WebSocket logger server (to send the log to the server instead of the browser console.log). The URL ‘logger’ parameter overloads the settings.

```
let c2c_config = {
   // Call
    call: 'JohnDoe',

    caller: 'Anonymous', // (single word)
    callerDN: 'Anonymous', // (words sequence).

    type: 'user_control',   // 'audio', 'video' or 'user_control'
    videoCheckboxDefault: false, // For 'user_control' call,
default value of video checkbox.

    videoSize: { width: '480px', height: '360px' },
    callAutoStart: 'no',  // 'yes', 'yes force', 'no'

    messageDisplayTime: 5,
    restoreCallMaxDelay: 20,

    allowCallWithoutMicrophone: true,
    networkPriority: undefined, // undefined,'high', 'medium',
'low', 'very-low'. Supported only in Chrome.


    // Optional test call to check line quality.
    testCallEnabled: true,
    testCallSBCScore: true,
    testCallUser: '5555',     // Call to this user for test call
    testCallAutoStart: false,
    testCallUseMicrophone: false, // Send microphone sound (true)
or generated tone/download sound (false).
    testCallVolume: 0.0,     // 1.0 .. 0.0. Hear or not test call
audio prompt received from SBC
    testCallMinDuration: 10,  // For SBC API request URL
"duration" value (converted to ms), for browser API minimum test
duration value.
    testCallMaxDuration: 20,  // For !testCallSBCScore
    testCallQualityText: {    // For testCallSBCScore only.
mapping SBC "color" voice quality score with corresponding text
message.
        'green': 'Good', 'yellow': 'Fair', 'red': 'Low', 'gray':
'N/A'
    },

    // Websocket keep alive.
    pingInterval: 10,    // 0 = off or interval in seconds.
    pongTimeout: true,
    timerThrottlingBestEffort: true,
    pongReport: 60,
    pongDist: false,

    // SDK modes.
    modes: {
        ice_timeout_fix: 2000,             // ICE gathering
timeout (milliseconds)
        chrome_rtp_timeout_fix: 13,        // Workaround of
https://bugs.chromium.org/p/chromium/issues/detail?id=982793
    }
};
```

4.1 Basic Settings

Use the following basic settings to configure your phone settings:

- **call**: (string) Call to user name (or phone number).

  Use value '_take_value_from_url_' to take this value from URL 'call' parameter.

  e.g.,: **https://example.com/c2c/html/?call=user17&server=sbc.example.com**

- **caller**: (string) Caller user name (one word according to SIP RFC 3261).
- **callerDN**: (String) Caller display name (words sequence).
- **type**: (String) Call type. 'audio', 'video' or 'user_control'
  - 'audio' audio only call
  - 'video' video call
  - 'user_control' audio or video call (selected by the user):
    - Call started as audio or video according to your video check box settings.
    - Camera can be switched on/off during call.
    - Video is received from the remote side (if received re-INVITE with video)
    - Mode can be set as audio only, one way or two-way video.

- **videoCheckboxDefault**: (boolean) Starts call as video (or audio) for 'user_control' call type.
- **videoSize**: (object) width and height of received video.
- **callAutoStart**: (string) Starts call automatically after page loading. 'yes', 'yes force' or 'no':
    - Not recommended.
    - It doesn't always work on all browsers and operating systems.
    - It may look like a violation of user privacy; the user has not decided anything yet. It only opens the HTML page, but the phone has already turned on the microphone and camera and started calling without any user interaction:
        - **'yes':** Start if audio autoplay policy is enabled
        - **'yes force':** Start always.
        - **'no':** Don't start call automatically
- **messageDisplayTime:** (number) A phone message is displayed during this time (seconds).
- **restoreCallMaxDelay**: (number) Time interval in seconds. If during an open call user navigates to another page on the same site and then returns to the phone page or makes the page reload, the call is restored if the time interval is less than the number.
- **allowCallWithoutMicrophone:** (boolean)
    - **False:** This is the normal setting. If the microphone is not set, the "No microphone or speaker" message appears. To use the microphone, the click-to-call phone user should connect the headset and reload the page.
    - **True:** This can be set for automatic testing or for a special case when the user calls from the PC without the microphone and wants to hear (if the speaker is present) and see the other side (if the other side sends a video and call type is "user_control"). In this case, the phone sends an audio player stream: "absenceMicrophoneSound" (see c2c_soundConfig).
- **networkPriority**: (string or undefined). This is only supported in Chrome.

    Possible values:

    - Undefined
    - 'high'
    - 'medium'
    - 'low'
    - 'very-low'

    Set sending RTP DSCP network priority:

    'Undefined' means don't change. When using IP differentiated services, the field value is 0x00.

4.2 (Optional) Test Call Settings

Some customers want to check their network connection quality between the phone and the SBC. The SBC is configured with a special Test Call username and special license. Multiple test calls are allowed simultaneously. The SBC automatically answers incoming test calls and plays a sound prompt. The SBC evaluates the sound quality score within 10-20 seconds, terminates the call and sends a SIP BYE with a special header and sound score.

Another option is to use the browser code to calculate sound quality. This method is not accurate as it only checks delays and lost RTP percentages. Code the browser sound quality rating located in voice_quality.js. The file can be deleted if you configured SBC sound quality rating.

- **testCallEnabled:** (boolean)

Shows the ‘test’ call button on the phone page and browser connection information.

- **testCallSBCScore:** (boolean)

  Tests the call voice quality score calculated by the SBC API (true) or by using browser API (false).

> ⓘ It is recommended you use the SBC voice quality score, because it works for all browsers. If you use the browser quality score, please uncomment in the *html/index.html voice_quality.js* file.

- **testCallUser**: (string)

  Call this SIP user for a test call.

- **testCallAutoStart:** (boolean)

  Start the test call automatically after page loading, when the autoplay policy enables ‘play sound’ or when a microphone is used for a test call.

- **testCallUseMicrophone:** (boolean)

  For a test call, send the sound from the microphone (true) or play a generated tone or the recorded sound (false).

- **testCallVolume:** (number, range: 0.0 .. 1.0)

  Listen (1.0) or not (0.0) to test the call voice prompt received from the SBC.

- (Optional) **testCallMinDuration:** (number in seconds)

  There are two possible values:

  - **Test call with SBC voice quality score** (recommended)

    Set testCallMinDuration (used in URL converted to milliseconds) in case the SBC terminates the call.

  - **Test call with browser voice quality score:**

    Set both testCallMinDuration and testCallMaxDuration in case the phone terminates the call.

  If the call receives remote-inbound-rtp (i.e., RTCP data that is used to calculate the browser score), the call is terminated after the minimum duration.

  If the data is not received, the test call continues until it receives the data or until the testCallMaxDuration interval has elapsed.

- **testCallMaxDuration:** (number seconds)

  For browser sound score only. The call always terminates after testCallMaxDuration interval.

- **testCallQualityText**: (object) for SBC sound quality score:

  Maps SBC ‘color’ quality score with a corresponding text message.

  The next block of parameters defines WebSocket keep-alive settings. You can use default values to define these settings.

> ⓘ The *ping* command is not the only command used to check whether the connection is live. You can also refresh the NAT connection.

For more information, refer to the setWebSocketKeepAlive method described in the *WebRTC Web Browser Client SDK API Reference Guide*.

5 Phone Customization

If you are a webmaster, you can customize this phone with the following:

- HTML file **html/index.html** (add/remove text, pictures, icons)
- CSS styles file **css/c2c.css** (modify styles)
- Javascript code file: **js/c2c.js**

5.1 JavaScript Code Modifications

This phone does not use any JavaScript GUI frameworks and is therefore very compact. It loads quickly even when using slow Internet (the demo size is 0.6 Mb).

Using large pictures or frameworks can increase its size by an order of magnitude (up to 10..20Mb).

The size can also be increased if the phone is running on different devices, for instance on desktops, tablets or mobile phones, and using different screen resolutions. make sure you use web responsive design.

5.2 GUI Settings for Different Phone States

The functions in the js/c2c.js file defines the GUI:

- **c2c_gui_phoneDisabled():** Disables phone if you are using a browser that does not support WebRTC API.
- **c2c_gui_phoneBeforeCall():** Phone before call waits until you press the **Call** or **Test** button.
- **c2c_gui_phoneCalling():** You pressed the **Call** button and phone is calling.
- **c2c_gui_phoneDuringCall():** Call is established.
- **c2c_gui_phoneOnRemoteHold():** The other side places the phone on hold during a all.

Fill out and send an arbitrary HTML form as an INVITE header. This is an optional feature.

To enable it:

- uncomment <form id=”user_order””> in file html/index.hmtl
- uncomment function c2c_create_x_header() In file js/c2c.js

Use the provided code as the feature demonstration.

International Headquarters
1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

AudioCodes Inc.
200 Cottontail Lane
Suite A101E
Somerset, NJ 08873
Tel: +1-732-469-0880
Fax: +1-732-469-2298

Contact us: https://www.audiocodes.com/corporate/offices-worldwide
Website: https://www.audiocodes.com

Document #: LTRT-14032