

## Using WebRTC for calling to ACCESS/BRIC-Link codecs from browsers

---

### What is WebRTC?

**WebRTC** is a new videoconferencing standard being added to web browsers. By design, without any other software installed, **WebRTC** can enable video calls between computers using browsers. In this way, it can emulate (and even out-perform) the functions of programs like Skype and Facetime.



Here's a secret: the "engine" to deliver **WebRTC** is already built into many browsers (maybe the one you're using now) and if you open a link to a **WebRTC** server, special javascript code can "wake it up", and allow you to make or accept calls from other **WebRTC** users.

### OK, it's a great video conferencing tool. How does it affect my codecs?

If you're not interested in **WebRTC's** video capabilities, you can turn them off. But when used for audio, the default encoding algorithm used provides excellent fidelity and low delay. When you use this to talk to your Comrex IP codec, you can make high-quality calls into your station from most web browsers!

### Tell me more about the audio format used.

The algorithm is called **Opus**, and has generated quite a lot of buzz in the IP telephone community. It has very low delay and rivals some of the best codecs in quality. And since it doesn't carry any licensing costs, it can be used in free applications like **WebRTC**. It delivers full bandwidth (20KHz) voice and music over data channels as low as 64 Kb/s.

### Sounds great! What are the limitations?

First of all, not all browsers support **WebRTC** yet. Currently, on Mac and Windows, the list includes Google Chrome, Mozilla Firefox, and Opera. On Android, Chrome and Firefox support it. Plug-ins are available for you die-hard Internet Explorer users as well.

### What about my iPhone?

Sorry, but Apple has chosen not to participate in **WebRTC** iOS support yet, and has restricted browser apps or plug-ins that try to support it. Hopefully they'll have a change of heart soon.

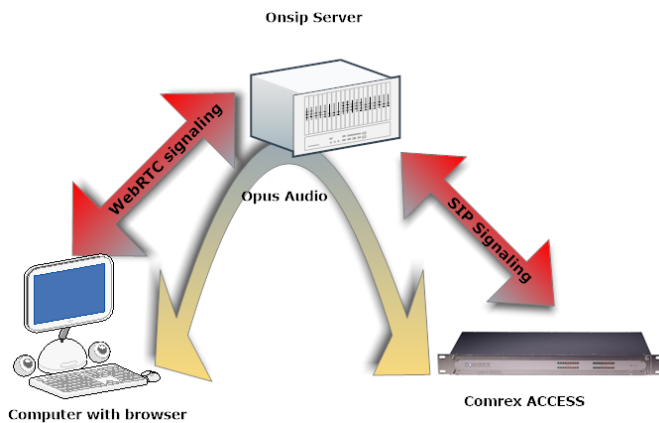
### So do I just make a WebRTC call to my codec?

Not quite. Comrex codecs don't natively speak **WebRTC**, but as of **firmware 3.0** they do speak **Opus**. Comrex codecs can encode, decode and deliver **Opus** over the SIP protocol (named **EBU 3326/SIP Settings** in the menus). And you can convert from **WebRTC-Opus** to **SIP-Opus** using free online conversion servers.

# Using WebRTC for calling to ACCESS/BRIC-Link codecs from browsers

## This sounds complicated. Explain.

It's not so bad. We're going to recommend use of a SIP provider called **Onsip.com**. They will provide you free credentials that you can key into your Comrex codec. They will also provide you a web link you can post or email to anyone. Once your hardware codec is registered with **Onsip**, anyone who clicks the link in a compatible browser can connect to your Comrex.



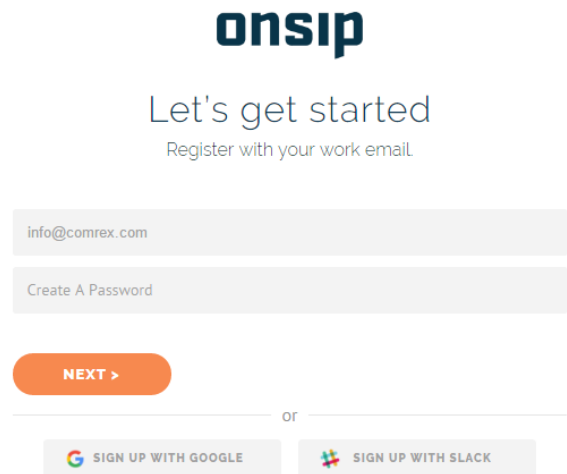
As shown above, **Onsip** handles the SIP portion of the call (to the codec) and the **WebRTC** portion of the call (to the browser) and does a translation of signaling. But the **Opus** audio flows through untouched and is encoded/decoded on each side.

## Can you take me through it?

You'll need:

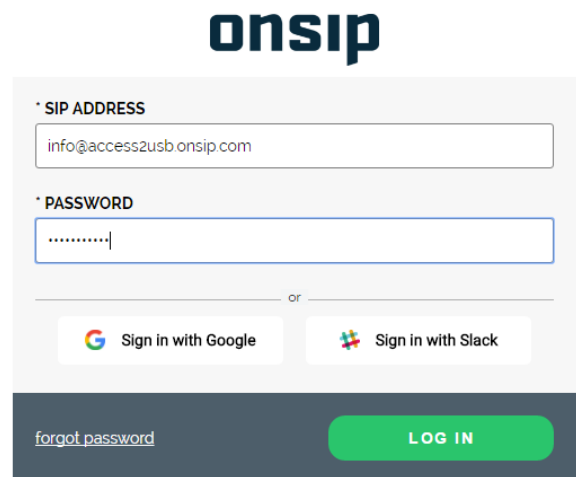
- 1) A computer with a microphone (built in is OK for testing, a USB headset will help performance a lot)
- 2) A Comrex BRIC-Link or ACCESS codec using **firmware 4.0p10** or higher

First, you will create a new account at **Onsip.com**. Click on the Start Free button in the top right. It will bring you to the following screen:



An email will be sent to you to activate your account. Once activated, a confirmation email will be sent. This will include a link to login to your account.

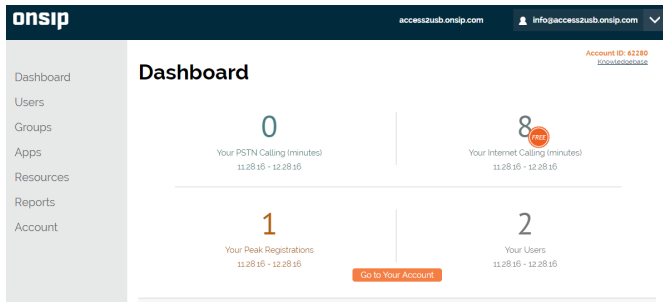
<https://admin.onsip.com/>



© 2004 - 2016 Junction Networks. All Rights Reserved.

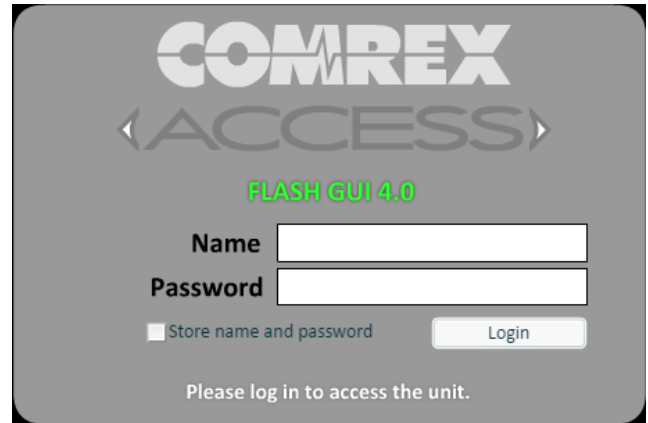
# Using WebRTC for calling to ACCESS/BRIC-Link codecs from browsers

Once logged in, you will be directed to a dashboard.



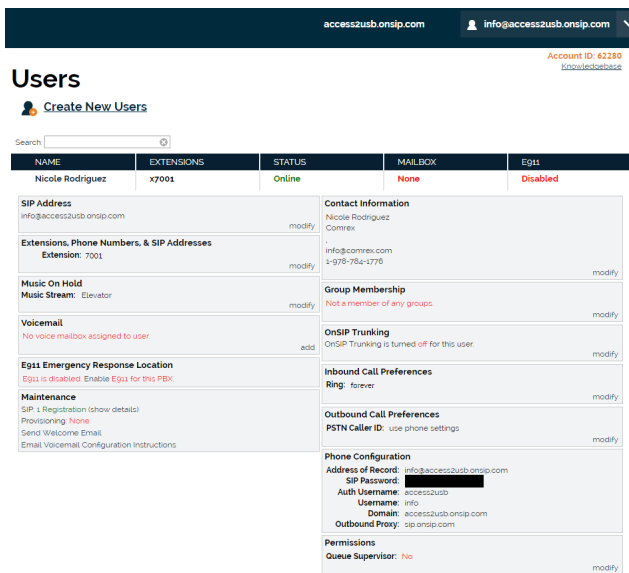
You can now log out of your account.

Now, log in to the web-based interface of your Comrex codec.

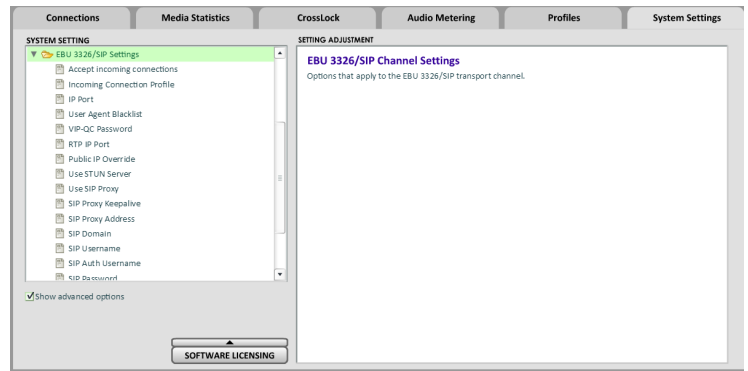


Click on **Users** in the menu to the left.

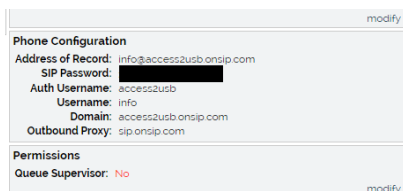
Select the username in the list to display user information below.



The changes we'll be making are on the **System Settings Tab**. Make sure the box labeled "Show Advanced Settings" is checked, and the **EBU 3326/SIP Settings** entries are expanded.



Copy the credentials under **Phone Configuration** and save it somewhere. This information is needed to configure your Comrex codec.



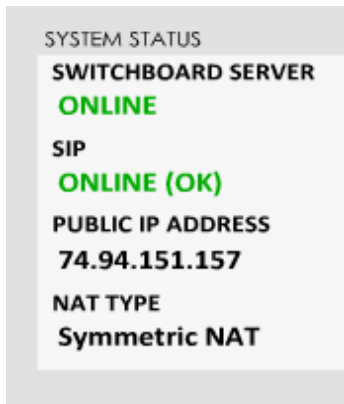
Make sure the following entries (and only these entries) are set correctly:

- 1) **Accept Incoming Connections**-> *Enabled*
- 2) **IP Port**-> *5060* (default)
- 3) **RTP IP Port**-> *6014* (default)
- 4) **Use SIP Proxy** -> *Enabled*

# Using WebRTC for calling to ACCESS/BRIC-Link codecs from browsers

- 5) **SIP Proxy Address** -> *Outbound Proxy*  
(from credentials, probably **sip.onsip.com**)
- 6) **SIP Domain** -> *Domain* (from credentials)
- 7) **SIP Username** -> *Username*  
(from credentials, probably your login name)
- 8) **SIP Auth Username** -> *Auth Username*  
(from credentials)
- 9) **SIP Password** -> *SIP Password*  
(from credentials)

If all these parameters are correct, navigate back to the connections tab and you should see an **ONLINE (OK)** status under **SIP**.



If any error messages appear here, carefully check the credentials as they were input. Also, be sure **UDP Port 5060** is not blocked for outgoing data. If the status shows **“Online”**, you’re ready to try a **WebRTC** call.

One more note about ports: The SIP traffic from your codec communicates to the server for signaling on outgoing port **UDP 5060**. This will usually work fine except in highly secure environments. But the incoming SIP media arrives on a different pair of ports. The default is **UDP 6014/6015**. If your codec

is behind a router, these ports are probably blocked to incoming traffic. There are several workarounds for this:

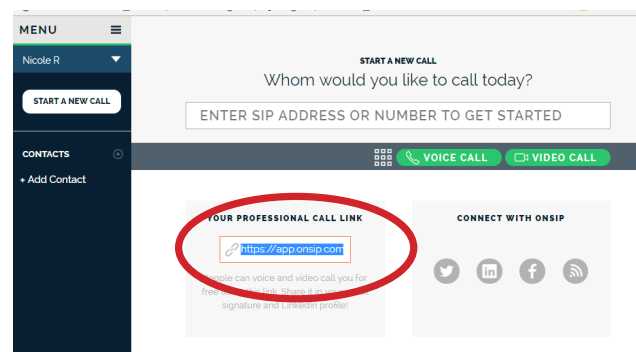
- 1) Your router may have a **SIP ALG** function in its firmware. In fact, it may be defaulted as active. If you can take incoming **WebRTC** calls using the procedure below (and you know your router blocks the ports) your **ALG** may be doing the work. If calls can’t be received, check your router for a **SIP ALG** option and turn it on.
- 2) You can program your router to forward **UDP 6014** and **6015** to your codec via a **“port forwarding rule”** in the router setup.
- 3) You can set your codec on a **DMZ** in the router setup, which will allow all unsolicited traffic to reach the codec.

Assuming your codec is registered with **Onsip**, you should know be able to make a SIP call.

Log in to your **Onsip** account with this link:

**<https://app.onsip.com/app/>**

On the dashboard, copy the link provided for people to make calls to you.

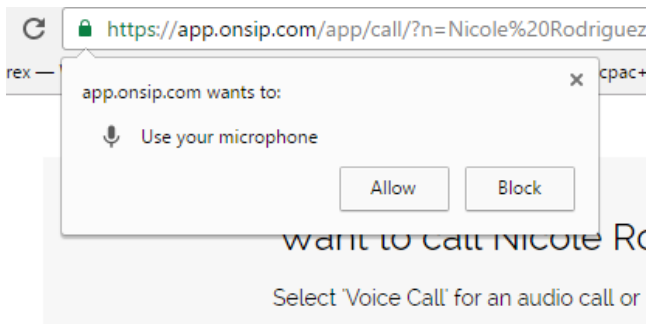


Copy and paste this link in a web browser.

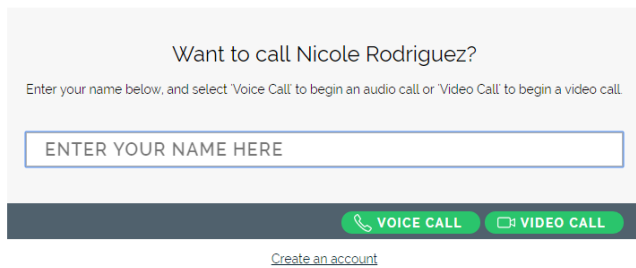
You’ll be presented with the calling screen, and the browser will ask for permission to use your

# Using WebRTC for calling to ACCESS/BRIC-Link codecs from browsers

microphone. (This may look differently on different browsers, but you must give permission for the call to work).



Key in a familiar name (this will appear on the incoming call name on the codec interface) and press the phone icon to make the call.



The call should be automatically answered by the codec and indicate an **Opus** audio connection in each direction. The call can be ended by either end at any time.

The image shows a screenshot of the COMREX ACCESS NICOLER interface. It has a header with "COMREX ACCESS NICOLER" and four tabs: "Connections", "Media Statistics", "CrossLock", and "Audio Metering". Below the tabs is a table titled "REMOTE ACCESS UNITS".

Name Profile	IP Address Crosslock Address	Current State Last State	Receive Status Transmit Status
VIP Guest - Aaron	199.7.173.101:5060	connected	Rx: Opus Mono Tx: Opus Mono
Loopback	127.0.0.1	not connected	

**Special note:** In this mode, your codec is open to accept SIP calls from anyone who knows your **Onsip** user name or your IP address. There exist SIP crawlers on the Internet looking for devices which may tie up your codec. When not using this function, you can disable it in the codec web interface via **System Settings->EBU3326/SIP Settings->Accept Incoming Connections->Not Enabled**