

# Turn Your Podcast Into A Interactive Live Streaming Experience



Joe Pruitt, 2009-25-09

The folks here on the DevCentral team have been producing a [weekly podcast](#) for a while now. Trying to keep our budget low, we opted to copy what some other podcasters were doing by making use of Skype for our audio communication and found a great little Skype add-on called [Pamela](#) which creates high-quality WAV files from Skype conversations.



We created a dedicated Skype account on an old machine here in the office that will auto-record whenever that account is added to a conversation. We would occasionally have differences in audio levels between the callers so we incorporated the awesome [The Levelator](#) (from the Conversations Network) into the post production and we were set.

A few weeks ago we broke through our one hundredth podcast and along with that milestone we decided to expand things a bit by making the podcast more interactive. I spent a few weeks investigating how to best accomplish this and I ultimately decided on a setup. I figured that I'd go ahead and share it with you all so you can see how we do it on our end. Here's the high-level diagram:



I experimented with a mixing board but ended up with a software only solution that anyone can implement. Everything besides the top left three items in the image were added to enable us to record our Skype podcast as well as stream it to UStream and allow callers with TalkShoe.

## The Components

Here's a list of the following components and how we used them.

**Skype** - <http://www.skype.com> - \$0

Skype, for those that don't know, is a software program that allows you to make video and voice calls across the internet. Calls are free to other skype users and you can purchase the ability to call out to land line numbers or to get your own number so you can accept phone calls.

We have a dedicated server with Skype and Pamela running to do the recording and each person on the podcast is connected through the main Skype instance on our publishing system.

#### **Pamela Skype Recorder - <http://www.pamela.biz/en/> - \$24**

Pamela is an add-on program for Skype that will allow you to record your Skype conversations. It features Skype call, video, and chat recording, answering machine, video mail, and skype based publishing. We use it primarily for the ability to automatically record skype calls to either our skype account or our Skype dial-in number. .WAV files are created and we are emailed when a recording is completed.

#### **Windows Media Player - \$0**

Windows Media Player is the default music/video player on the Windows platforms. We use it to play background music into our skype and UStream sessions. You could just as easily use another audio player as long as it supports the ability to customize which output audio device it uses.

#### **Virtual Audio Cable - <http://software.muzychenko.net/eng/vac.html> - \$30**

This little gem really saved the day with our setup. There are various Mac programs that allow you to map audio streams from one program to another one but this is the best one I could find for Windows. Virtual Audio Cable allows you to transfer audio (wave) streams between applications and/or devices. In our setup, it allows us to map windows media player back into Skype and our multiple audio streams back into our live video processing through VidBlaster.

#### **VidBlaster - <http://vidblaster.com/> - \$0**

VidBlaster is a powerful, economic way to record, stream, and produce high quality videos. VidBlaster comes in three versions: Home, Pro, and Studio with the only difference being the number of "modules" you can use at one time. What's best about this product is that you can use it for free if you can live with a "VidBlaster" add in the top right corner of your video stream. We'll likely put the couple hundred bucks down for the Pro version but as of now it hasn't costed us a cent.

#### **Adobe Flash Media Encoder - <http://www.adobe.com/products/flashmediaserver/flashmediaencoder/> - \$0**

The Adobe Flash Media Encoder allows you to capture live audio and video while streaming it in real time to a flash media server. While we could stream directly from VidBlaster to UStream, the quality is not as good as moving some of the processing down to the client.

#### **UStream - <http://www.ustream.tv> - \$0**

UStream is the live interactive video broadcast platform that enables anyone with a camera (or VidBlaster!) and an internet connection to quickly and easily broadcast to a global audience of unlimited size. Best of all, it's free!

#### **TalkShoe - <http://www.talkshoe.com> - \$0**

We struggled for a while trying to figure out the best way to include a "live" audience in our recording. TalkShoe is a service that enables anyone to easily create, join, or listen to live interactive discussions, conversations, podcasts, and audioblogs. It fit the bill of just what we needed and since we already had our Skype conversation going, bolting on TalkShoe was as simple as adding the conference number to our group Skype conversation. TalkShoe also has some pretty nifty user controls allowing the moderator to control muting of each of the participants. Another great free service!

## Production Setup Walkthrough

Here's the steps I go through in preparation for our weekly podcast.

1. Create 3 Virtual Audio Cables with the Virtual Audio Cable Control Panel (VAC #1, VAC #2, VAC #3)
2. Start up 2 instances of the Virtual Audio Cable Audio Repeater. Set the first one (R1) from VAC #3 to VAC #1 and the second from VAC #3 to VAC #2.
3. Start and login to Skype. Go into the Audio settings and change the audio input to VAC #1 and the output to VAC #2.
4. Start Windows Media Player and change the speaker device settings to output to VAC #3.
5. Start VidBlaster and setup various screen captures, set the video resolution to 640x480 and the frame rate to 15fps (much higher than that drives the CPU way up). Finally click "Start" on the Streamer module.
6. Start the Adobe Flash Media Encoder. Load the stream configuration that you can download from your UStream show's advanced settings and then Select VidBlaster for the Video Device and "Line 2 (Virtual Audio Cable)" (ie. VAC #2) for the Audio device. Click "Start" to begin streaming.
7. Login to our UStream.tv account and click "Broadcast Now" for your show. Another browser window will come up detecting the media stream. Click "Start Broadcast" to start the stream and click "Start Record" to begin recording your stream on the server.
8. Bring up the Skype window and create a conference with the podcast members including our account that auto-records the conversations. Then finally, call the TalkShoe conference bridge and initiate the meeting on their side.
9. Login to the TalkShoe account on their website and click the option to join the meeting. When the admin console is loaded, you can optionally select "Record" from the top left to have TalkShoe make an alternate recording.

At this point we are recording our Skype session in audio, UStream is recording the video stream and TalkShoe is recording a secondary audio recording. At this point, I cue up the intro music in Windows Media Player and start into the podcast.

When we are finished, I reverse the process above by stopping and saving the various recordings. I take our Pamela based recording, run it through The Levelator, convert it to a mp3 with Audacity, edit the ID3 tags, and publish it to our media server. At this point, the blog post is created and we are done for the week.

## Reflections

There are a few things that are still causing some issues. The main issue is that of horsepower on my media system. The desktop I'm using for all of this was not meant to run these types of CPU intensive applications. While it works, my dual-core CPU system is hovering at 99-100% CPU usage during the podcast which is a bit worrisome. Depending on the success of the podcast, we may invest in a new desktop to run this on. But, seeing how we've spent well under \$100 for the entire software suite, I don't think we'll have much trouble justifying it!

Hopefully this helps some of you out there with some ideas and I'd love to hear feedback on how I could do things better!

-Joe

---

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](http://f5.com)

F5 Networks, Inc.  
Corporate Headquarters  
[info@f5.com](mailto:info@f5.com)

F5 Networks  
Asia-Pacific  
[apacinfo@f5.com](mailto:apacinfo@f5.com)

F5 Networks Ltd.  
Europe/Middle-East/Africa  
[emeainfo@f5.com](mailto:emeainfo@f5.com)

F5 Networks  
Japan K.K.  
[f5j-info@f5.com](mailto:f5j-info@f5.com)

---

©2016 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at [f5.com](http://f5.com). Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113