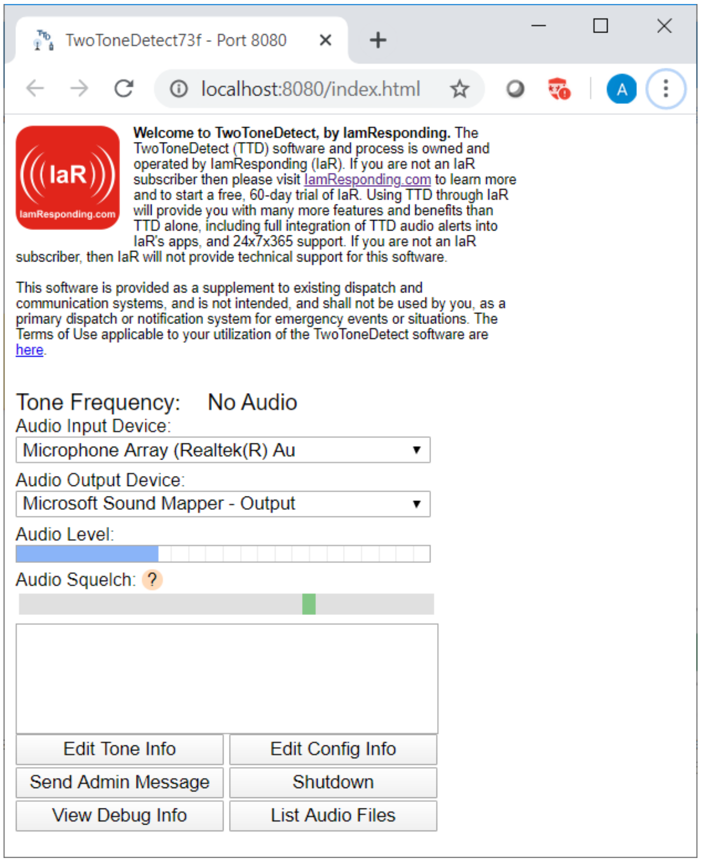
TwoToneDetect Setup Guide

Contents

What is TwoToneDetect?

TwoToneDetect by IamResponding is a program that detects two-tone pages, records audio, and sends the recording to email or cell phones. Versions are available for both Windows and a Raspberry Pi running Raspbian Linux.



Welcome to TwoToneDetect, by IamResponding. The TwoToneDetect (TTD) software and process is owned and operated by IamResponding (IaR). If you are not an IaR subscriber then please visit IamResponding.com to learn more and to start a free, 60-day trial of IaR. Using TTD through IaR will provide you with many more features and benefits than TTD alone, including full integration of TTD audio alerts into IaR's apps, and 24x7x365 support. If you are not an IaR subscriber, then IaR will not provide technical support for this software.

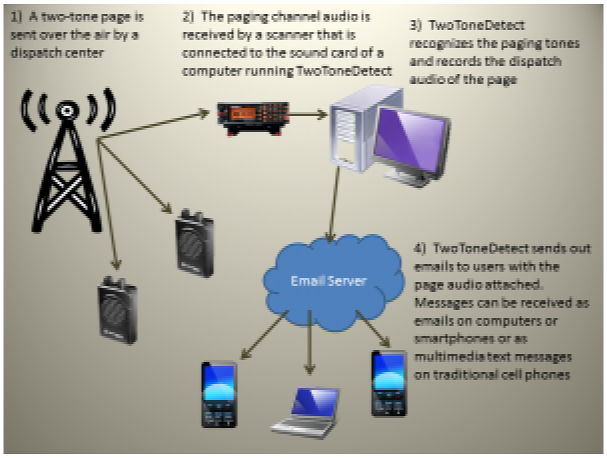
This software is provided as a supplement to existing dispatch and communication systems, and is not intended, and shall not be used by you, as a primary dispatch or notification system for emergency events or situations. The Terms of Use applicable to your utilization of the TwoToneDetect software are here.

Tone Frequency: No Audio

Audio Input Device: Microphone Array (Realtek(R) Au

Audio Output Device: Microsoft Sound Mapper - Output

TwoToneDetect interfaces with a radio receiver or scanner. It uses the computer's sound card to listen to the received audio and detect two-tone sequential paging sequences (Quick-Call 2 and similar formats). This type of paging is commonly used for fire department alerting. Upon detecting known two-tone pages, the program will record dispatch audio and email that audio to a specified email address. Multiple tone sets and email addresses are supported.


1) A two-tone page is sent over the air by a dispatch center
2) The paging channel audio is received by a scanner that is connected to the sound card of a computer running TwoToneDetect
3) TwoToneDetect recognizes the paging tones and records the dispatch audio of the page
Email Server
4) TwoToneDetect sends out emails to users with the page audio attached. Messages can be received as emails on computers or smartphones or as multimedia text messages on traditional cell phones

Program Features

- Can be fully integrated into the IamResponding.com service to allow users to receive notifications via the IamResponding phone app
- Decodes virtually unlimited numbers of user-specified two-tone and long-tone pages
- Handles "stacked pages", including trimming stacked tones from audio recording
- Upon detecting a valid tone set, records user-specified length of audio and emails it to a specified address as a .mp3 or .amr file attachment (requires user download of ffmpeg.exe)
- Near real-time playback of recorded audio, providing "pager-like" functionality with scanner and PC
- Can be used to trigger external programs or scripts upon tone detection
- Audio frequency counter to assist in finding tone frequencies with PC sound card
- Works in parallel with audio streaming software
- Support for uploading audio files to an FTP server

Disclaimer

This program should never be used as a primary alerting method for mission-critical communications. It is meant as a secondary notification method only. Delays in email or text messaging systems can be minutes or hours long. Do not use this program to replace pagers, only to supplement them!

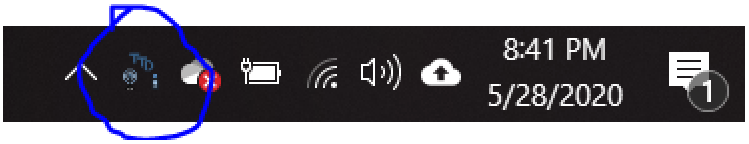
IamResponding Integration

If you plan to use TwoToneDetect with IamResponding.com, please follow the IaR Quick Start guide.

Windows Download & Installation

After downloading the .zip file, simply unzip into the folder of your choice on your computer (make sure that you put it in a location on the computer that you have file write access to). There is no installer. Inside the folder there are two executable files (.exe files). They are “TwoToneDetect##.exe” and “TwoToneDetect##_debug.exe”. The only difference between the two programs is that the _debug version of the program will open up a “debug” screen in addition to the main program screen. This can be helpful to see what’s going on “behind the scenes” of the program if you encounter any problems.

Once TwoToneDetect is running you should see an icon in the Windows taskbar:



Click on the taskbar icon to open the TwoToneDetect GUI in a web browser. Alternatively, you can browse to this location to access the GUI: http://localhost:8080/index.html

Follow the steps in the [Program Configuration (Windows and Pi)]() portion of this guide to finish setting up TwoToneDetect.

Raspberry Pi Installation Using Image

Raspberry PI Image with TTD already installed on it can be found [here.]()

The image file includes the full Raspberry PI operating system with TwoToneDetect preinstalled and configured to automatically start on boot.

We’ve tested the image successfully with both a Raspberry Pi 3 Model B and Raspberry Pi 4 Model B 2GB version. The image is designed to work with a Raspberry PI 3 & 4, older Raspberry PI hardware such as the 1, 2, Zero & Zero W are much more likely to run slower or have issues due to the older specs.

Checklist:

- Raspberry PI 4 (1GB or 2GB version recommended)
- Micro HDMI to HDMI cable
- Raspberry PI enclosure
- USB-C power supply
- USB sound card
- Micro SD card (16GB or larger)

- USB to micro SD card adapter
- Radio/Scanner (use a battery eliminator if using a radio)
- Audio cable

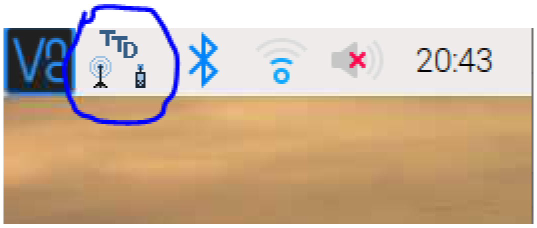
Image installation instructions can be found on the Raspberry PI website.

This tutorial follows the Win32DiskImager steps at the following link:
https://www.raspberrypi.org/documentation/installation/installing-images/windows.md

(Image installation instructions from Raspberrypi.org website)

- Insert the SD card into your SD card reader. You can use the SD card slot if you have one, or an SD adapter in a USB port. Note the drive letter assigned to the SD card. You can see the drive letter in the left hand column of Windows Explorer, for example G:
- Download the Win32DiskImager utility from the Sourceforge Project page as an installer file, and run it to install the software.
- Run the Win32DiskImager utility from your desktop or menu.
- Select the image file you extracted earlier.
- In the device box, select the drive letter of the SD card. Be careful to select the correct drive: if you choose the wrong drive you could destroy the data on your computer's hard disk! If you are using an SD card slot in your computer, and can't see the drive in the Win32DiskImager window, try using an external SD adapter.
- Click 'Write' and wait for the write to complete.
- Exit the imager and eject the SD card.
- Put SD card into Pi
- Plug in USB sound card prior to powering up Pi
- Power up Pi with keyboard, mouse, and monitor attached (optionally, SSH and VNC access are also enabled for advanced users who know how to use those tools and know how to find the IP address of the Pi on their network - default password is iamresponding)
- Change time zone to local (raspi-config via GUI)
- Change password (optional - default is iamresponding)
- Setup WiFi if necessary. Hardwired ethernet preferred for reliability reasons.

TTD will automatically start on boot (may take a minute or two). An icon should be visible in the system tray once it's running:



- Click on the taskbar icon to open the TwoToneDetect GUI in a web browser. Alternatively, you can browse to this location to access the GUI: http://localhost:8080/index.html
- Follow the steps in the Program Configuration (Windows and Pi) portion of this guide to finish setting up TwoToneDetect.

Raspberry Pi Manual Installation

This section describes how to get TwoToneDetect running on a $35 Raspberry Pi single board computer. Along with a $5 USB sound card and a cheap receiver ($50 Wouxon or similar), you can have a full TTD setup running 24/7 that consumes very little power for under $100.

Note: This guide was written for Raspbian Buster.

What you'll need

- A Raspberry Pi
- An SD card (4 GB or greater)
- 5V Power supply with a micro-USB cord to power the Pi
- A USB sound card (I use this one)
- A radio receiver (radio, scanner, pager, etc.)
- A network connection (wired ethernet recommended)

Getting Started

First, follow the instructions here to get the latest Raspbian operating system installed on your SD card. Follow the 'Download and image Raspbian directly' section from that page (about halfway down) for getting your image.

Next, take a look at this tutorial to get the Pi configured during the initial boot up. You'll want to configure the Localization Options (Locale, TimeZone, Keyboard, and Wifi country) and also Interfacing Options (enable SSH and VNC if so desired for remote access). These are available under Preferences/RPi Configuration.

Installing Prerequisites for TwoToneDetect

From a terminal, run the following to install pulseaudio:

```
 sudo apt-get install pulseaudio
```

Next run this to install a graphical audio control application:

```
sudo apt-get install pavucontrol
```

Then install ffmpeg, which is used to convert raw audio recordings to compressed MP3 and AMR files. To get AMR support we'll need to build ffmpeg from source (this will take awhile):

```
cd ~
git clone git://source.ffmpeg.org/ffmpeg.git ffmpeg
cd ffmpeg;
sudo apt-get install libopencore-amrnb-dev
sudo apt-get install libmp3lame-dev
sudo apt-get install libfdk-aac-dev
sudo apt-get install libass-dev
sudo apt-get install libopus-dev
sudo apt-get install libpulse-dev
./configure --pkg-config-flags=--static --bindir=/home/pi/bin
--enable-gpl --enable-libass --enable-nonfree --enable-libmp3lame
--enable-libopencore-amrnb --enable-version3 --enable-libopus
make
sudo make install
```

Finally install some required audio codecs:

```
sudo apt-get install libavcodec-extra58
```

Installing TwoToneDetect

Now we're ready to get the TwoToneDetect files. First let's make a TTD directory to put our files in, and then go into that directory:

```
mkdir TTD
cd TTD
```

Now we'll fetch the TwoToneDetect files.

```
curl -O http://www.twotonedetect.net/downloads/TwoToneDetect73f.tgz
```

And then unpack the files:

```
tar zxf TwoToneDetect73f.tgz
```

There should now be a number of files in your TTD directory. TTD can now be run as follows from a command prompt:

```
/home/pi/TTD/TwoToneDetect73f
```

Setting up pulseaudio

Unlike under Windows, we won't be able to specify your USB sound card directly as the audio input to TwoToneDetect. This is because the Pi's audio driver system doesn't support different sampling rates for the sound card. It will only allow us to sample at the default rate. Instead, we'll use pulseaudio as the input to TwoToneDetect, since pulseaudio will perform resampling from the USB sound card to TwoToneDetect. To set our audio input levels, we can use the pulse audio control GUI from the desktop. Access PAVucontrol from the Pi/Sound and Video option from the taskbar. Click on the "Input" tab and set the volume of your USB sound card appropriately, using your radio with open squelch or tuned to a weather broadcast channel to get the levels correct. Set any other inputs to zero volume.

Configuring TwoToneDetect

Running Multiple TwoToneDetect Instances

Run this command to list the input devices available to pulseaudio:

```
LANG=C pactl list | grep -A2 'Source #' | grep 'Name: ' | cut -d" " -f2
```

It will return something like this (exact device names will be different):

```
alsa_output.usb-0d8c_C-Media_USB_Headphone_Set-00-Set.analog-stereo.monitor

alsa_input.usb-0d8c_C-Media_USB_Headphone_Set-00-Set.analog-mono
```

```
alsa_output.usb-0d8c_Generic_USB_Audio_Device-00-Device.analog-stereo.monitor

alsa_input.usb-0d8c_Generic_USB_Audio_Device-00-Device.analog-mono
```

Next, when you start TTD, set the PULSE_SOURCE environment variable to whichever input you want to use from the step above (don't use the ones that end in "monitor"). Here is an example:

```
PULSE_SOURCE=alsa_input.usb-0d8c_C-Media_USB_Headphone_Set-00.analog-mono ./TwoToneDetect73f
```

So, to run two instances of TTD, you'd have two TTD folders set up, and use a different PULSE_SOURCE when starting each instance. You can use the same strategy with the PULSE_SOURCE environment variable to run different instances of darkice for streaming. Of course this can all be scripted for automation.

Automatically starting, monitoring, and restarting TwoToneDetect

The example scripts in this section can be used to automatically start TwoToneDetect when the Raspberry Pi boots up and to continuously check to make sure TwoToneDetect is still running and restart it if it crashes. There is more than one way to accomplish this...this is how I do it, but others have had success with other methods.

First, install the screen utility:

```
sudo apt-get install screen
```

Next create a script called TTDcheck.sh and make it executable. This script will check once per minute to see if TwoToneDetect is running. If it finds that it is not running, it will launch it. Contents should be something like this:

```
#!/bin/bash
while true
do
      ps auxw | grep TwoToneDetect | grep -v grep |grep -v SCREEN >>
/dev/null
      if [ $? != 0 ]
      then
            cd /home/pi/TTD
            screen -dmS ttd-screen ./startTTD.sh & > /dev/null
      fi
      /bin/sleep 60
done
```

Now create a script called startTTD.sh in your /home/pi/TTD directory, make it executable, and make the contents something like this:

```
#!/bin/bash
cd /home/pi/TTD/
/bin/sleep 3
./TwoToneDetect73f
```

Or if you want to specify a specific sound card to use when starting TwoToneDetect, startTTD.sh could look something like this:

```
#!/bin/bash
cd /home/pi/TTD/
/bin/sleep 3
PULSE_SOURCE=alsa_input.usb-0d8c_C-Media_USB_Headphone_Set-00.analog-m
ono ./TwoToneDetect71c
```

Finally, edit your /etc/xdg/lxsession/LXDE-pi/autostart file and add a line to start the /home/pi/TTD/TTDcheck.sh script on startup

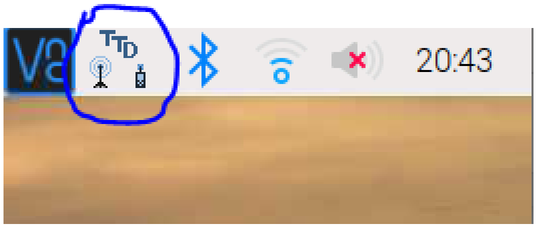
Optionally, you may also want to restart the Pi every night just to keep things running smoothly. Do do that run this command:

```
sudo crontab -e
```

And add this line to the file to reboot every night at 2 a.m. :

```
0 2 * * * /sbin/shutdown -r +2
```

One TwoToneDetect is running you should see this icon in the taskbar:



Click on the taskbar icon to open the TwoToneDetect GUI in a web browser. Alternatively, you can browse to this location to access the GUI: http://localhost:8080/index.html

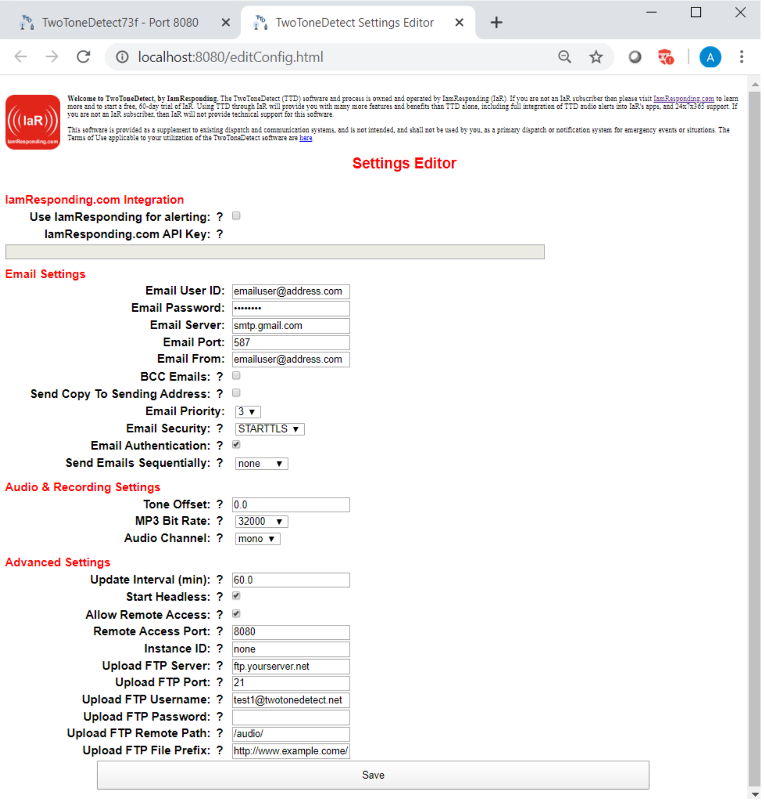
Follow the steps in the [Program Configuration (Windows and Pi)]() portion of this guide to finish setting up TwoToneDetect.

Program Configuration (Windows and Pi):

1. Select the audio input and output devices you want to use from the main GUI (Windows only)
2. Set the Audio Squelch level using the slider on the main GUI. The audio level should be below the squelch level ("No Audio" displayed) when there is no radio traffic present, and the audio level should be above the squelch level when there is radio traffic present.
3. Click on the "Edit Config Info" button to set up the configuration items according to your needs. See "Starting and Configuring the Program" section below
4. Click on the "Edit Tone Info" button to set up your tone sets and distribution lists. See the "Editing Tone Information" section below for more information. You can also manually edit the tones.cfg file using a text editor if you prefer.
5. Exit the program using the "Shutdown" button
6. Restart TwoToneDetect

Starting and Configuring the Program

1. Start TwoToneDetect
2. Select the Audio Input and Output devices from the drop down menus. It's best to use a line-in for the scanner input, but a microphone in can also work if set up properly in the Windows sound mixer. Use Windows sound mixer to adjust the input audio level, if needed. **Note:** On some newer Windows machines, an audio jack must be plugged in to the audio port before Windows will recognize the audio device. Not having anything plugged in to your sound card may prevent TwoToneDetect from starting.
3. Adjust the audio squelch level so that the audio level is under the threshold when no scanner audio is present, but above the threshold when scanner audio is present. You may find that tuning the scanner to a local weather station can help quickly find an appropriate squelch level. This can also help you adjust the sound levels in the Windows sound mixer.
4. Next, click the "Edit Config Info" button. The following menu should appear:


TwoToneDetect73f - Port 8080
TwoToneDetect Settings Editor
localhost:8080/editConfig.html
IaR
IamResponding.com

Welcome to TwoToneDetect, by IamResponding. The TwoToneDetect (TTD) software and process is owned and operated by IamResponding (IaR). If you are not an IaR subscriber then please visit IamResponding.com to learn more and to start a free, 60-day trial of IaR. Using TTD through IaR will provide you with many more features and benefits than TTD alone, including full integration of TTD audio alerts into IaR's apps, and 24x7x365 support. If you are not an IaR subscriber, then IaR will not provide technical support for this software.

This software is provided as a supplement to existing dispatch and communication systems, and is not intended, and shall not be used by you, as a primary dispatch or notification system for emergency events or situations. The Terms of Use applicable to your utilization of the TwoToneDetect software are here.

Settings Editor

IamResponding.com Integration

Use IamResponding for alerting: ? ☐

IamResponding.com API Key: ?

Email Settings

Email User ID: emailuser@address.com

Email Password: ••••••••

Email Server: smtp.gmail.com

Email Port: 587

Email From: emailuser@address.com

BCC Emails: ? ☐

Send Copy To Sending Address: ? ☐

Email Priority: 3

Email Security: ? STARTTLS

Email Authentication: ? ☑

Send Emails Sequentially: ? none

Audio & Recording Settings

Tone Offset: ? 0.0

MP3 Bit Rate: ? 32000

Audio Channel: ? mono

Advanced Settings

Update Interval (min): ? 60.0

Start Headless: ? ☑

Allow Remote Access: ? ☑

Remote Access Port: ? 8080

Instance ID: ? none

Upload FTP Server: ? ftp.yourserver.net

Upload FTP Port: ? 21

Upload FTP Username: ? test1@twotonedetect.net

Upload FTP Password: ?

Upload FTP Remote Path: ? /audio/

Upload FTP File Prefix: ? http://www.example.come/

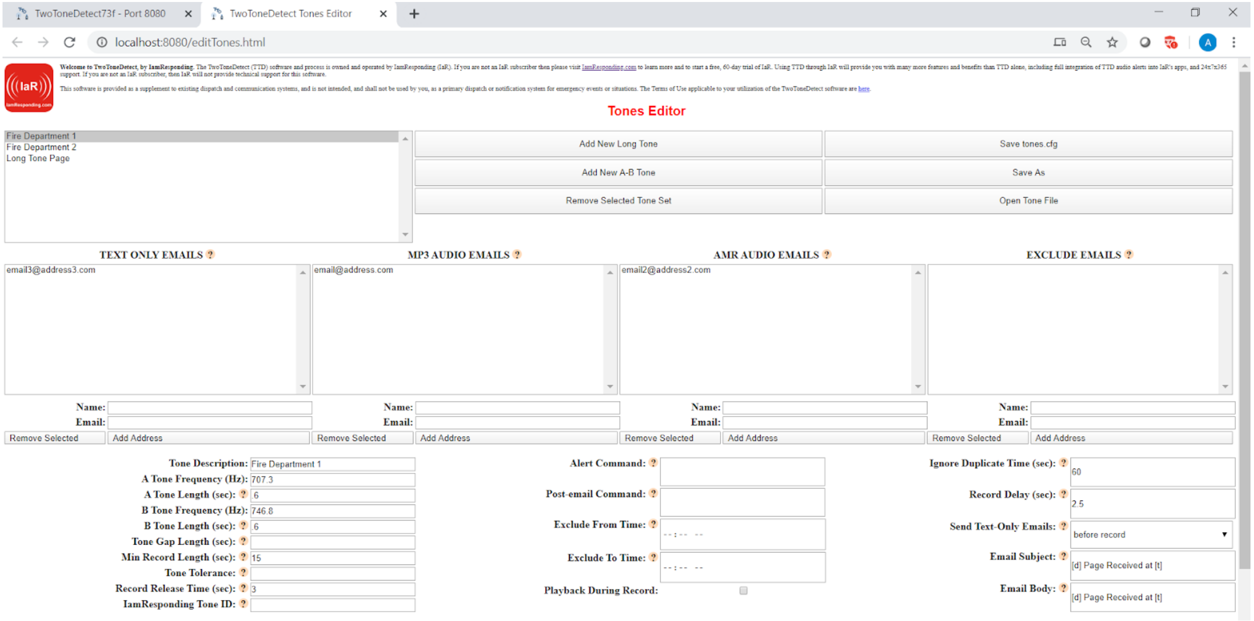
Configure the menu items as appropriate. Here is a brief description of each of the fields:

- **Email UserID**: UserID of the email account that you will be using to send emails FROM
- **Email Password**: Password of the email account that you will be using to send emails FROM
- **Email Server**: Address of the email server that you will be using to send emails FROM
- **Email Port**: Port of the email server that you will be using to send emails FROM
- **Email From:** Specify the address that you want to use in the "From" field when sending emails. Often the same as the Email User ID
- **BCC Emails:** When checked, emails will be sent as a blind carbon copy (BCC) so that email recipients don't see the email addresses of other recipients.
- **Send Copy To Sending Address:** When checked, a copy of the email will be sent to the address that it's being sent from. Some users have found that this helps prevent messages getting marked as spam in some scenarios.
- **Email Priority:** Sets the email priority for the message header [X-Priority]. 1 = highest, 3 = normal, 5 = lowest
- **Email Security:** Email security type as specified by your email provider. Typically set to "SSL" if using port 465 for email and set to "STARTTLS" if using port 587 for email.
- **Email Authentication:** Check this to send emails using authentication. Typically must be checked unless you're using your own in-house email server.

- **Send Emails Sequentially:** Selections available to send individual emails to recipients rather than a single message to multiple recipients. Doing this may help avoid email spam filters but may take longer to send.
- **Min Record Length**: Minimum number of seconds of audio that will be recorded after a valid page is received
- **Tone Offset**: This is a “fudge factor” offset (in percentage) that is applied to all incoming audio signals. This can be used to correct for sound card inaccuracy due to clock offset or drift.
- **MP3 Encoding Bitrate**: Select the desired mp3 encoding bitrate from the drop down menu. Lower bitrates result in smaller files, higher bitrates result in larger, higher quality files.
- **Record Release Time**: Number of seconds of silence necessary before recording will stop (dynamic record time). Max is 60 seconds, default is 0 (which makes recording 100% based Record Seconds parameter)
- **Audio Channel:** Specify which audio channel of the sound card to use. Typically set to "mono" unless you are connecting different radio scanners to each channel of your sound card for streaming purposes.
- **Update Interval:** Number of minutes between reloading email address information from the tones.cfg file
- **Start Headless:** Select this option to start TwoToneDetect "headless" without a GUI. TwoToneDetect can still be accessed via web browser on the local machine while running, even in "headless" mode.
- **Allow Remote Access:** Select this option to allow other computers on the Local Area Network to access the TwoToneDetect GUI via web browser.
- **Remote Access Port:** Specify the port to use for HTTP access to the TwoToneDetect GUI. For example, if Port 8080 is selected, you can access the GUI by pointing a web browser to http://ipaddress:8080/index.html
- **Instance ID:** Specify text two display in the title bar of the GUI to help differentiate multiple instances of the program running on the same computer.
- **Upload FTP Server:** FTP server to upload audio files to after recording
- **Upload FTP Port:** Port to use to access the FTP server (usually Port 21)
- **Upload FTP Username:** Username to access the FTP server to upload audio files to
- **Upload FTP Password:** Password to access the FTP server to upload audio files to
- **Upload FTP Remote Path:** Folder or path on the remote FTP server to upload audio files to. For example, to upload files to the /audio/ subdirectory, put /audio/ in this field.
- **Upload FTP File Prefix:** HTTP path at which the audio files can be accessed by a browser after uploading. Used when sending Pushover notifications. For example: http://www.myserver.com/audio/

Editing Tone Information

In order to tell the program what tone sets we want it to look for and act upon, we'll use the Tones Editor GUI that can be accessed by clicking the "Edit Tone Info" button on the main page of the program:



Fire Department 1
Fire Department 2
Long Tone Page

TEXT ONLY EMAILS: email3@address3.com
MP3 AUDIO EMAILS: email@address.com
AMR AUDIO EMAILS: email2@address2.com

Tone Description: Fire Department 1
A Tone Frequency (Hz): 707.3
A Tone Length (sec): .6
B Tone Frequency (Hz): 746.8
B Tone Length (sec): .6
Tone Gap Length (sec):
Min Record Length (sec): 15
Tone Tolerance:
Record Release Time (sec): 3
IamResponding Tone ID:

Alert Command:
Post-email Command:
Exclude From Time: --:-- --
Exclude To Time: --:-- --
Playback During Record: ☐

Ignore Duplicate Time (sec): 60
Record Delay (sec): 2.5
Send Text-Only Emails: before record
Email Subject: [d] Page Received at [t]
Email Body: [d] Page Received at [t]

Using this GUI editor you can add new A-B tones or Long tones, and edit existing tones. Be sure to click the "Save to tones.cfg" button before selecting a new tone to edit or changes will not be saved. The sections below explain what the different fields mean. If a parameter is optional, it can be left blank.

Mandatory Parameters for Two-Tone (A-B format) pages:

- **Atone** – Frequency (in Hz) of the first (A) tone in the two-tone sequence
- **Btone** – Frequency (in Hz) of the second (B) tone in the two-tone sequence
- **Atonelength -** Duration of A tone (in seconds) that must be detected before the program will alert.
- **Btonelength** – Duration of B tone (in seconds) that must be detected before the program will alert

Note: Setting Atonelength and Btonelength both to .6 should be a good starting point for most common two-tone paging formats. Adjust as necessary.

- **Description** – Description of the tone set. Avoid using special characters such as slashes, quotation marks, or punctuation

- **mp3_Emails** – This is a list of email addresses that the program will send mp3 attachments to whenever a valid page is received for this tone set. To send to multiple email addresses, simply separate them with commas.

Mandatory Parameters for Long Tone pages:

- **Longtone** – Frequency (in Hz) of the tone
- **Longtonelength** – Length of the time the program must see the long tone before alerting. Set to a value shorter than or equal to the actual tone duration.
- **Description** – Description of the tone. Avoid using special characters such as slashes, quotation marks, or punctuation
- **mp3_Emails** – This is a list of email addresses that the program will send mp3 attachments to whenever a valid page is received for this tone set.

Additional Optional Parameters for either page type:

- **Tone Tolerance**: Tolerance (in percentage: setting of 0.02 = 2%) of the tone detection algorithm. As an example, if a tone of 1000 Hz is specified and the tolerance is 2%, the program will detect tones from 980 Hz to 1020 Hz. If this is set too narrow some tones may be missed. If it is set too wide, false detections may occur. A value of 1-2% is usually sufficient.
- **amr_Emails** – This is a list of email addresses that the program will send AMR format audio file attachments to whenever a valid page is received for this tone set. AMR format audio files can be played by most cellular phones when received as a text or multimedia message (MMS).
- **text_Emails** - This is a list of email addresses that the program will send a text-only email to whenever a valid page is received for this tone set. The email will not contain any information, it will just let the recipient know that a page has been received. This email will be sent immediately upon tone detection, prior to audio recording
- **record_delay** - Number of seconds that the program will wait to begin recording after a valid tone set is found
- **ignore_after** – Number of seconds that the program will ignore a tone set after it has been received. This is useful if a dispatch center normally sends multiple pages for the same call. It can be used to prevent multiple email or text messages from being sent for the same call.
- **alert_command** – This parameter can be used to point to a windows Batch file (.bat) or other executable file. The file that is specified here will be executed whenever the specified tone set has been detected but before the audio has been recorded. As an example, a batch file could be created that plays a pre-recorded audio file when the page is received.
- **post_email_command** – This parameter can be used to point to a windows Batch file (.bat) or other executable file. The file that is specified here will be executed whenever the specified tone set has been detected and after the audio has been recorded.

Variable substitution will insert the current Tone Description in place of [d], the mp3 filename in place of [mp3], the AMR filename in place of [amr], and the WAV filename in place of [wav] (all paths relative to program directory).

- **exclude_from, exclude_to, and exclude_emails** parameters – These three parameters are used together to prevent emails from being sent to certain addresses when a page is received during a certain time of the day. This can be used to prevent emails from being sent during daily test pages, or to only send pages for certain duty shifts. When these three parameters are used, the tone sets will still be detected, but if the time of day of the page falls between exclude_from and exclude_to, the email will not be sent to the addresses in the exlude_emails parameter. Multiple email addresses can be specified by separating them with commas. This field will prevent emails from being sent to addresses in both the mp3_emails and amr_emails parameters.
- **playback_during_record -** If set to 1, the program will play the page audio through the computer speakers (or other audio device as specified on the main program screen) while it is being recorded. If set to 0, it will not play the audio while recording.

A note on sending pages to cell phones:

To send the page audio to a cellular phone, you must send the email to the carrier's Multimedia Messaging Service (MMS) email address, not the normal SMS (text messaging) email address. For some carriers these two email addresses are the same, for others, they are different. For example, to send a normal text message to a Verizon phone via email you would address it to ##########@vtext.com, but to send a MMS message (containing a picture or audio file), you would need to address it to ##########@vzwpix.com. A list of SMS and MMS gateways for different wireless carriers can be found here.

Program Operation

TwoToneDetect continuously checks the sound card input for audio. If the audio level exceeds a set threshold, the program measures the most dominant tone in the audio signal (even if it is not a true "tone"). If it sees a valid two tone or long tone page, it will start an alert thread that will record audio after the page, convert the recorded audio into AMR and MP3 format, and then send the audio as an email attachment. The program can also trigger external programs or scripts to run upon tone set detection.

FTP Support

Starting with v70, the the following fields in the config.cfg file (or the "Edit Config Info" menu on the main TwoToneDetect screen) are can be used to have TwoToneDetect upload a recorded MP3 file to an FTP server:

```
upload_ftp_server =
upload_ftp_port =
upload_ftp_username =
upload_ftp_password =
upload_file_prefix =
```

The upload_file_prefix should be set to the web storage location where the audio files can be accessed via http. For example: http://www.mywebsite/audiofiles/

Sending a Hyperlink via SMS

To send a hyperlink to the web address of the file that was uploaded via FTP, set the "Email Body" field in the TonesEditor program to the root web address, and use the [mp3] tag to insert the filename of the file that was uploaded to the web server. Note that the [mp3] tag includes the relative path to the file. For example:

Setting "Email Body" to: http://www.myfd.org/[mp3]

will send this hyperlink in the body of the message:

http://www.myfd.org/audio/Tone1_2016_12_22_15_32_17.mp3

Editing the tones.cfg File Remotely

There are several ways that the tones.cfg file can be managed remotely. This allows the program to be running on a computer in one location while new email addresses can be added remotely.

IamResponding.com Integration

By integrating TwoToneDetect with IamResponding.com, all tone information and user notifications can be managed from any location using the IamResponding web interface.

LAN GUI Access

Starting with v71, the TwoToneDetect GUI can be accessed from anywhere on your Local Area Network via web browser by turning on the "Allow Remote Access" configuration setting. When this is turned on, the GUI can be accessed via web browser by browsing to index.html at the IP address of the computer that TwoToneDetect is running on and the appropriate port. For example: http://192.168.1.13:8080/index.html

Note: Because there is no login required to access the GUI, it is recommended that you NOT open this port to allow access from outside your LAN. If you want to access the GUI from outside the LAN, use a VPN to gain access to your LAN and then browse to the GUI from within the LAN.

Hosting the tones.cfg file remotely

TwoToneDetect is able to fetch a tones.cfg file that is hosted remotely and accessible via http.

To do this, create a file called remote.cfg with a text editor and place it in the same directory as the TwoToneDetect program (you can use the remote_example.cfg file included with the program as an example). The contents of the remote.cfg file should look like this:

```
[REMOTE]
URL = http://www.website.com/tones.cfg
proxy = 8.8.8.8
key = 12345ABC
```

The first two lines in this file are mandatory. The first line should not be changed. In the second line, put the URL where your tones.cfg file is stored. The next two lines are optional, delete them if you don't need them. If you need to access the web through a proxy server, put that information in the third line. If not, delete the entire line. The fourth line is an optional 8-character DES encryption key. If desired, the remotely-stored tones.cfg file can be encrypted using DES-ECB encryption to prevent anyone from seeing the information. If the "key" line in the remote.cfg file is present, the program will use the specified 8-character key to decrypt the remotely stored file when loading it. If the remote file isn't encrypted, delete the fourth line.

If the program is not able to retrieve the remotely hosted file for any reason, it will revert to using the local tones.cfg file that is in the same directory as the program. If it is able to retrieve the remotely hosted file, it will overwrite the local tones.cfg file with the information that was retrieved remotely and this new information will then be stored locally.

Using A Cloud Drive Service

Another option is to use a cloud drive service such as Dropbox or Google Drive. To do this, install the cloud drive software on the computer running TwoToneDetect. This will create a folder that you can then put the TwoToneDetect program folder into. Since this folder is "in the cloud", you can access the tones.cfg file remotely and edit it.

Redundant Operation with Multiple Instances

IamResponding.com Integration

When TwoToneDetect is set up for integration with IamResponding.com, multiple instances running at different locations can be used to ensure that notifications are sent to responders even if one instance goes down due to a power or internet connectivity outage.

FTP Redundancy

Multiple instances of TwoToneDetect running in different locations can work together to provide a redundant backup alerting system.

For example, two copies of TwoToneDetect can be running at fire stations opposite sides of a county or district where they are on different power grids and have different internet service providers. If the power or internet goes out at the primary location, the secondary location will detect that the primary location is down and will send the emails instead.

Or, two copies of TwoToneDetect can be running on the same computer with different email servers specified in their configurations. If the primary email server is down and fails to send, the secondary copy of TwoToneDetect will send the emails using the secondary server instead.

How it Works

Both instances of TwoToneDetect must have access to a common file that they can both read and write to. This file can have any name, but in these examples we'll call it CommonToneLogFile.txt. When TwoToneDetect is configured for redundant operation and detects a valid tone set, it will attempt to read CommonToneLogFile.txt to see if any other instance of TwoToneDetect has already detected the same tone and sent emails. If it finds that no other instance of the program has already sent emails for the detected done set, it will send the emails and then update CommonToneLogFile.txt so that other instances of the program will not send email.

In order to ensure that multiple instances of the program are not trying to read or write to CommonToneLogFile.txt at the same time, each instance of the program is configured with a delay. The delay for the primary instance of TwoToneDetect should be set to zero, while each additional instance should be configured with a delay of around 15-60 seconds to allow the primary instance to write to the file before the secondary reads the file.

TwoToneDetect supports the ftp protocol for remote file access. If both instances of TwoToneDetect are running on the same computer (for email server redundancy), a locally stored file can be used.

If the remote file is accessed via ftp, a second "common" text file is used by each instance of the program to check to see whether the other instances of the program are online. In this way, if the email or power goes out at one location running TwoToneDetect, the other location will detect this and send an email to an administrator so that they are aware of the outage. In these examples we'll call this file CommonStatusFile.txt

To configure TwoToneDetect to look for this shared file, create a text file called redundant.cfg in the program directory. This file will have the following format:

```
[REDUNDANT]
file_access = ftp
tone_tracking_file = CommonToneLogFile.txt
redundant_delay = 30
redundant_instance_alias = Backup at Station 2
ftp_server = ftp.twotonedetect.net
ftp_port = 21
ftp_username = ftp_username
ftp_password = ftp_password
TTD_tracking_file = CommonStatusFile.txt
redundant_down_alert_time = 400
redundant_admin_email = email@address.com
```

Explanation of file parameters:

- **file_access:** options are **ftp** or **local**.
- **tone_tracking_file:** Name of the file used to track detected tone sets across multiple running instances of the program. For local files, specify the entire path to the file i.e. c:\path\to\filename.txt. For a common file stored on an ftp server, specify only the filename i.e. filename.txt.
- **redundant_delay:** Number of seconds to wait before checking the common file. For the primary instance of TwoToneDetect, set this to 0. For the secondary/backup instance of TwoToneDetect, set this to a reasonable delay. 30 seconds is suggested as a starting point.
- **redundant_instance_alias:** This field allows you to specify an alias for this instance of TwoToneDetect for easier identification when emails are received. For example, you could set it to something like "Primary", "Backup", "Station 1 Radio Room", "Tower Site", etc.
- **ftp_server:** If accessing a remote common file via ftp, specify the server location here. If accessing a local common file, this line can be omitted.
- **ftp_port:** If accessing a remote common file via ftp, specify the ftp port here. Default is port 21. If accessing a local common file, this line can be omitted.
- **ftp_username:** If accessing a remote common file via ftp, specify the ftp username here. If accessing a local common file, this line can be omitted.
- **ftp_password:** If accessing a remote common file via ftp, specify the ftp password here. If accessing a local common file, this line can be omitted.
- **TTD_tracking_file:** Name of the file used to track whether other instances of the program are online. This parameter is only used if file_access = ftp.
- **redundant_down_alert_time:** Amount of time to wait before sending an email to notify an administrator that one of the instances of TwoToneDetect is offline. This parameter is only used if file_access = ftp.

- **redundant_admin_email:** Email address to send an alert message to if an instance of TwoToneDetect is detected as being offline. This parameter is only used if file_access = ftp.

Tips

1. When setting a system up for redundant operation, it is probably a good idea to take advantage of TwoToneDetect's remote tones.cfg capabilities to ensure that both copies of the program are accessing the same remotely-hosted tones.cfg file. This will ensure that both the primary and backup instances of the program are using the same list of tones and users at all times.
2. Make sure both the primary and redundant system are having their clocks automatically updated via NTP. Most modern operating systems do this automatically, but double check to make sure it's enabled on your system. It's important that the clocks of the two systems are synchronized regularly.
3. If you reboot the systems regularly (this is recommended), have the systems reboot at different times of day so that one is always online. Having the reboot times staggered also helps reduce the changes that both instances are always accessing the remote files at the same time.
4. Differences in performance on the two systems can lead to different response times. It may take one system longer to convert the WAV files to MP3/AMR format than the other, and it may take one system longer to send emails than the other. Keep this in mind when choosing the redundant_delay time interval. The longer this delay time is, the less likely it is that you will get duplicate emails for the same tone set. On the other hand longer delay times will result in the message being delayed longer when the primary system is down.

A Word of Caution

While cloud drive services like Dropbox and Google Drive may be an option for hosting the common file (using the "local" option in the file_access parameter of the redundant.cfg file), the synchronization of files between remote computers is left to the whims of the cloud drive's application. Because TwoToneDetect redundant operation is time-sensitive, cloud-drive solutions may not perform well, and could result in multiple emails being sent for the same tone set. Your mileage may vary.

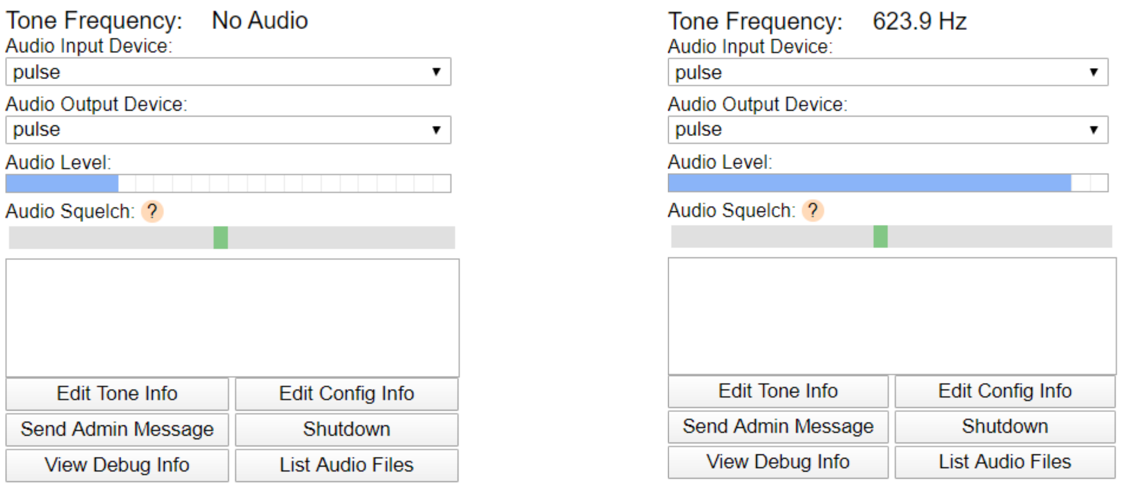
FAQ

How do I find my tone frequencies?

The easiest way is to ask your dispatch center. Alternatively, here is a pdf document that shows how to use Audacity to find tone frequencies from recorded audio files.

My audio recording is longer than I expect. What can I do to fix this?

This is typically caused by the Audio Squelch Threshold being set too low. The slider on the mains screen of the program is used to adjust the Audio Squelch Threshold. The slider should be adjusted so that the main screen of the program displays "No Audio" when there is no radio traffic present, but shows audio frequencies when there is radio traffic present. After adjusting the slider, make sure to exit the program using the "Exit" button on the main screen to save the changes.



Tone Frequency: No Audio
Audio Input Device: pulse
Audio Output Device: pulse

Tone Frequency: 623.9 Hz
Audio Input Device: pulse
Audio Output Device: pulse

Image on the left shows what the GUI should look like with no radio traffic. Image on the right shows what the GUI should look like when there is radio traffic.

Audio recording works as follows:

1. After a tone set is detected, the TwoToneDetect will wait the amount of time specified by the Record Delay parameter in the Tones Editor for the tone set in question
2. After the Record Delay has expired, recording begins
3. Recording continues for the amount of time specified by the Record Time parameter in the Edit Config Info menu
4. After the Record Time expires, the program continues to record until no more audio is detected for the amount of time specified by the Record Release Time parameter in the Edit Config Info menu.

If the Audio Squelch Level is set too low, the program will continue to detect audio even when there is no radio traffic present, and the recording will continue until a 60 second timeout occurs after the Record Time expires. If your audio recordings are about 60 seconds plus your Record Time, you probably need to adjust your Audio Squelch Level up a bit.

I’m having problems with Gmail, what should I do?

The most reliable way to address this is to set up TwoToneDetect to send alerts as push notifications through Iamresponding.com

When using Gmail as your email provider to send messages from TwoToneDetect, make sure you use your full email address in the Email UserID field. For example: you'll want to use *username@gmail.com*, not just *username.*

Also, if you use two-step authentication, you'll want to set up an application-specific password that TwoToneDetect can use with your Gmail account to prevent two-step authentication issues. For details on how to set this up, see this page.

If you don't use two-step authentication, you might need to allow "less secure" apps to access your Gmail account.

Some users have reported that Gmail sometimes blocks their messages as spam, and they have to contact Google to get the account unblocked. Users have also found that it helps to have all of the email and MMS addresses that you're sending to in your Google "Contacts" to help reduce the chance that it becomes marked as spam by Google. As an alternative, Yahoo mail has been tested to work with TTD, and the email service of your internet service provider may be another option.

My cell phone carrier is blocking TTD messages. What should I do?

1. The most reliable way to address this is to set up TwoToneDetect to send alerts as push notifications through Iamresponding.com
2. Have customers contact their cellular provider’s customer service and notify them of the problems. At first, they will likely blame the problem on the user's device. Be sure to notify them that all users in your agency using their carrier are experiencing the problem.
3. If you are using the "BCC" option in TwoToneDetect, try disabling it. Some users have reported that doing this has increased their success rate with some carriers.
4. Consider using SMS notifications with a hyperlink to the audio file rather than MMS messages with an audio attachment.
5. If you are using a free email service like Gmail to send messages, consider switching to an email account hosted by your department. If your department has a web page, changes are you can set up an email account through your department's domain for free.
6. Consider sending the alerts via email rather than MMS. The user can set up their phone to provide a specific notification when an email is received from a specific address or with a specific subject. Instructions on how to do this for various smartphone operating systems can be found here.