# unrealaudio

| COLLABORATORS | | | |
|---|---|---|---|
| | *TITLE* :<br><br>unrealaudio | | |
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | | November 11, 2024 | |

| REVISION HISTORY | | | |
|---|---|---|---|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

# Contents

# Chapter 1

# unrealaudio

## 1.1 unrealaudio.guide

```
unrealaudio  v0.2

Dodgy docs and dodgy program written by  Michael Cheng
```



## 1.2 Introduction

```
'unrealaudio' is a hack I put together over one weekend that will enable the
amiga to play back audio samples in real time over the internet.

It is *NOT* a RealAudio player.  RealAudio, IPhone, TrueSpeech etc etc all use
proprietary algorithms, and Lucifer will be skating to work the day that any of
these get released in native amiga versions.  So I decided to get off my
backside and see if we could compromise.

'unrealaudio' uses a public domain codec called GSM 06.10 [the full GSMToast
archive for the amiga is available from aminet util/pack/GSMToast.lha] which is
used in the area of cellular telephony.  It is also used for voice coding samples
for real time play back from the internet.  GSM isn't used as widely as RealAudio,
nor is it of the same quality, but for the time being it's all we have.

*NOTE* the GSM untoast binaries included are slightly modified.  The -a option
now outputs a raw byte and not 8bit A-law
```

I did consider making an ADPCM implementation that would work in my hack but
    a) I couldn't compile the amiga version (I don't have SAS/C)
    b) I didn't want to play with the SAS source to get it to output in a
        way that would be useful
    c) No one uses ADPCM encoded samples on the net for real time playback
        (kind of makes the whole thing pointless)

The following document outlines my methodology and I have compiled the
voice codec for the amiga. Anyone who can think of a better way of getting the ↩
   data
from the net to the codec, or from the codec to the audio device is welcome to
mail me. If you think my implementation sucks and there are much better ways to
do it, then tell me. Tell the world. Do it yourself and upload it to aminet.
Whatever. Just get it done.

## 1.3 History

v0.2
    No longer require amisox (Thanks to Chris Masto)
        untoast is now slightly changed from the util/pack/GSMToast release
        in that the -a option outputs raw audio (not A-law) and this can be
        fed directly to the AUD: device

v0.1
    First release.

## 1.4 Terminology

    codec - coder/decoder   compressor/decompressor
    FIFO  - first in, first out

## 1.5 Requirements

    - an amiga
        Both a 68030/68881 and a 68000 binary are included.
        I can't vouch that you'll get real time decompression on anything less
            than a 25Mhz 030 though. Experiment and get back to me if you can.

    - a net connection. A 14.4k connection *might* just squeeze in as usuable
        but I doubt it. A 28.8k connect should be good. And an ethernet connect
        should be great. [It is of course possible to download the compressed
        sample on a 14.4k modem and play it back, but that's not the aim of the
        game, in this case]

    - The following programs will need to be installed correctly
        (all can be found on aminet)

        + ixemul43+          dev/gcc/ixemul
        + FIFO-handler       util/misc/fifolib38_1.lha

```
        + audio-handler     mus/play/Audio-Handler.lha

    - The following programs come in this archive.

        + untoast       this is the codec
                        030/881 & 68000 versions included with this archive
                        [amiga GSMToast codec archive avail at  util/pack/GSMToast ↩
                           .lha]

        + web           This program grabs URL's using the CLI
                        [type 'web' at prompt to see syntax]


    - make sure web, untoast are in your path and that the fifo- and
        audio-handlers are enabled.
```

## 1.6 Methodology

```
From the above requirements, the more astute reader will have already figured
out what I am about to do.  In short, I read a URL using web into a FIFO pipe.
The other end of that pipe feeds into the codec. The codec outputs raw audio
data to the audio-handler.

Steps:

1. Open a CLI and type

    untoast -a -c <fifo:inc/r aud:buffer0x8000 <return>

This should sit there and do nothing.  It outputs data to aud:, but
it has yet to receive any data in from fifo:inc

2. Open a second CLI and type

web http://town.hall.org/Archives/radio/IMS/Unformatted/uf_22.au.gsm fifo:inc/wmKe

This starts grabbing a compressed sample file from town.hall.org and feeds it
into the pipe fifo:inc.


You should now be the proud listener of real time audio from the net.


3. To stop the audio:
    Ctrl-C in the untoast cli
    Ctrl-C in the web cli
```

## 1.7 Sources of GSM Audio

```
To get a list of all the GSM samples that are out there, use your web browser to
cruise over to
                http://www.cam.org/~noelbou/gsm_wine.html
```

## 1.8 Problems

- There is a periodic pause

This is a problem with audio-handler. The output is double buffered, but when it swaps buffers, there is a pause. For small size buffers (0x8000) the pause is tiny, but frequent. For large buffers (0x40000 is the largest) the pause is huge (3seconds??) but there's a longer time between pauses.

- If you hear just big pauses and little snippets of sound

You need a minimum 1.6k/s connection to the site from which you are receiving the audio.

- Could still do with some speed improvements

If you read the docs with GSMToast, I explain that there is a bug with the -F flag for untoast. It *should* make decompression 2x faster, with a little bit of signal degradation. Instead, it just makes the decompression about 6x longer. :(

- Only got a 14.4k connection?

This codec was built around compressing 8000Hz samples. It gets a compression ratio of about 4.8:1 meaning about 1.6k/s and this means that you are *just* going to miss out with only a 14.4k modem. You may be lucky, and your modem compression may bring you the required throughput. If you do have a 14.4k connection, try killing off all other net connections (ftp, irc etc). If you do get it working, drop me a line and I'll amend this bit of the docs.

## 1.9 Future

The Future is in your hands. If you feel you can implement any of the following or want to contribute to the program so it does, don't hesitate to contact me.

Possibilities

- implement GSM as a MIME type

- instead of redirecting untoast output to AUD: make it use the audio.device

- Fix audio-handler to get rid of that pause

- FIFO allows you to copy breaks (eg ctrl-c) directly into the pipe.
  Implement this as a way of stopping the audio

- The use of 'web' to stream data to an application independant of the web browser can have other uses.

  + playback mpegs over the net
  + progressive display of pictures by redirecting into a viewer
  + animation playback over the net
  + I've even redirected 22khz aiff samples over a fast link

## 1.10 MIME type

GSM encoded files are of mime type audio/x-gsm having a suffix of gsd, or gsm. The gsd suffix indicates a text file which contains data in the format:

```
GSM_URI|http://full.address/path/file.gsm
```

and thus points to the actual sound file.

Put together a script file which is launched when you encounter a .gsd file that will extract the url, grab the file, setup the pipes, play the file and clean up after itself.

## 1.11 Author

Mail me, Mail me, Mail me. Send me ideas, money, chocolate or a life.

Michael Cheng
memfc@alinga.newcastle.edu.au
http://joffre.newcastle.edu.au/
Cstar on #amiga

## 1.12 Acknowledgements

This little project wouldn't have been possible without the following people who write such useful stuff for the amiga.

Nick Loman (zeus@mistral.co.uk) for Web
[ rauper on #amiga Author of Zeus BBS soft ]
[ http://www.mistral.co.uk/zeus/ ]
Matt Dillon et al for FIFO
Martin Brenner (martin@ego.oche.de) for Audio-Handler
Dana Cooper (dgc3@midway.uchicago.edu) for AmiSOX

Thanks to ideas from

Christopher Masto (mastoc@rpi.edu)
Pointed out that it's easier to get untoast to output 8bit raw in the format we want rather than use amisox.

..and to Fastlane and Mayday on #amiga for doing the initial testing.

## 1.13 Appendix

Some good sources for speech stuff

ftp://teltec.ucd.ie/pub/speech

```
http://www.cs.tu-berlin.de/~jutta/   (NB may be down permanently?)
http://deskfish.cs.titech.ac.jp:8001/squish/
```