

MAKEUEF V2.3

Welcome to MakeUEF, an accurate data format computer cassette preservation program.

A second programmer has extended MakeUEF from V0.3b. MakeUEF now comprises of an accurate version for professional cassettes and a simpler version for BBC standard files on amateur cassettes that has the file name MakeUEFam.exe. MakeUEF V2.3 was released in October 2008. This manual mainly addresses the professional cassette version of MakeUEF.

The professional cassette version will attempt to convert genuine BBC/Electron/Atom and any other KCS cassette recordings to cassette images via CSW sound files. All data waves, baud rates and gap lengths are now recorded with great accuracy.

The amateur cassette version supports more input formats (CSW, WAV or RAW format or your sound card) and has a wider wavelength toleration range to allow easier reading of cassettes. Also the UEF file output is simpler with standard baud rate and carrier tone lengths. It only supports standard BBC files in the 1200 baud rate format.

IMPORTANT NOTICES

- Before November 2004, all UEF cassette images available on the Internet will have been created with the earlier 0.x releases of MakeUEF – hence they only contain program data. The gap lengths, baud rates etc. will have not been retained.
- As cassettes can now be preserved to a highly accurate level, please create/recreate UEF (and CSW) files from your original cassettes and send them to the Acorn Preservation project at <http://www.acornpreservation.org>. Files created by the new, 1.x and 2.x versions of MakeUEF will have a secondary extension of HQ (meaning High Quality), to distinguish them from the older files.

Try to also send the LOG files created while making UEF files and small sized scans of the cassettes of both sides in order to ascertain what cassette was used.

N.B. the Stairway to Hell web site – <http://www.stairwaytohell.com> – will continue to collect and archive low quality BBC/Electron cassette images that have been created with MakeUEF v0.3b (or MakeUEFam).

- If submitting a UEF file try to not lose its accuracy by for instance: removing or editing chunks in UEF files, converting sound sample files to a different sampling rate, and using MP3 or any other loss inducing format.

HOW TO USE

THE RECORDING PHASE:

44100 samples per second is the minimum sampling rate required by MakeUEF and 22050 for the Amateur version. Higher sampling rates are ok and will produce CSW files no larger than lower rates.

Refer to the document 'How to record WAV files for conversion to CSW.PDF' that is supplied with CSW Viewer.

THE CSW FILE CREATION PHASE:

Refer to the document 'How to record WAV files for conversion to CSW.PDF' that is supplied with CSW Viewer.

THE UEF FILE CREATION PHASE:

Start a command prompt window by using the Run... option on the start menu and typing CMD. MakeUEF only runs on Windows at present.

Run the program with no switches or the -h switch to get a list of options.

If you haven't ran MakeUEF before use the -g switch explained in the detailed descriptions of command line options section.

CSW is the recommended format to use with both versions of MakeUEF. Run MakeUEF with, for example, this instruction:

`makeuef -i 747 (Doctorsoft) (B) (Tape) [side-lab].csw`

Type the correct cases of letters in file names. MakeUEF should speed through the CSW file creating a UEF file. Listen for a beep, which indicates an error and use the pause key and another, such as the space bar to observe the output. Also a LOG file will be produced that the output can be viewed with.

If you see no data blocks, consider whether you used the correct phase shift settings. Some cassettes require a change of phase shift at a gap between files, particularly compilation cassettes. MakeUEF will expect the initial phase shift to be 0 degrees. The phase shift can be set with the -w switch. Many cassettes require the command: -w 0 180. MakeUEF will report "Phase shift changed to +0.", if -w 0 180 is not used because the UEF format has a default phase shift of 180.

If you see any bad checksums or misread blocks you can try running MakeUEF again with -r switch. The -r switch can be used to identify a point where reading is failing. Some cassettes have a non-standard data format or a non-standard block format, which would require switches explained later. In the worst case you may have to record again with different tone/volume settings and make a new CSW file. Control-C or Control-Break can be used to terminate MakeUEF when you know it isn't working as expected.

MakeUEF will automatically set the expected baud rate to a mean baud rate it calculates from the whole CSW file for file 1 and onwards. This value is reported.

Sometimes files have a mean baud rate that is quite different to a whole cassettes mean baud rate. If this is so you should rerun MakeUEF with the -s switch and the mean baud rate reported for those files. You don't have to be particularly accurate choosing the mean values. Even if all the data blocks were read this switch will help MakeUEF to get waves at the starts and ends of files.

Always look for hiss or sound spread from the other side of a tape being read as readable waves. There is more about this in the limitations section.

I have found the average cassette is converted easily. Firebird cassettes tend to be difficult ones and often have non-standard data blocks. MicroPower cassettes sometimes have a large gap before the first file on the unlabeled side.

Your motherboard speaker has to be working to hear a beep.

MakeUEF is not designed for putting together a UEF file from multiple files.

DIRECT SOUND INPUT TO MAKEUEF AMATEUR

Start MakeUEF Amateur with a command such as `makeuefam -o mycassette`. Play the cassette and watch for all the blocks being read. When all the blocks are read, press Escape or Q. Otherwise stop the program with Control-C or Control-Break.

Refer to the document 'How to record WAV files for conversion to CSW.PDF' that is supplied with CSW Viewer for information on recognising phase shift. A phase shift of 0 does not require a switch. A phase shift of +90 requires -l. A phase shift of +180 requires -p. A phase shift of +270 requires -l and -p. CSW files should have only a phase shift of +0 or +180 and only the -p switch is effective.

It might also help to use the -s switch. Particularly difficult files will be read better by following the CSW file creation process.

The sampling rate is found automatically for sound card input. See the hardware requirements section for more information.

Turn off loudness buttons on amplifiers. Turn off Dolby noise reduction on cassette players.

READING ATOM CASSETTES

Atom cassettes are normally in the 300 baud rate format so you will want to use -y 1 300. If a first file on a cassette were a standard Atom file you would want the switch -z 1 ATM.

Standard Atom files have a carrier tone followed by a header, then carrier tone, then data and finish with a check byte. The check byte is created from all the bytes of the header and the data blocks. A gap follows the check byte, which makes it slightly different to other bytes, which are followed by one short wave. MakeUEF reads all this without the need for any further switches. MakeUEF sees a header and its associated data block, as 2 data blocks.

Acornsoft Atom cassettes often have a strange 'identification' file at the end without a leading carrier tone. I think this might be the Acorn System 1 format. The -u switch can be used to read these. The first wave of these blocks is often not clear in a CSW file because of the omitted carrier tone. Using the -d with MakeUEFam may make the first wave clearer for reading.

At the end of Bug-Byte cassettes there often is a series of singular bytes amongst carrier tone in the 8N1 format. This will be read better with the switch -z block number 8N1 although it is not necessary to change from ATM.

FILE NAMING CONVENTIONS

The Acorn Preservation project has information on naming files.

DETAILED DESCRIPTION OF COMMAND LINE OPTIONS

- g <signature>

Adds the given message to the UEF file. The message will be recorded in the Windows registry and written to all later files until this option is used without a message. I recommend keeping it brief with just your name and your cassette player model and the usual message for the preservation site. This is important if you are submitting UEF files to the www.acornpreservation.org site. Some cassette players negate the polarity of the signal, which is a reason for specifying what the model is. The www.acornpreservation.org site would like to see 'Created for www.acornpreservation.org.' in the signature.

This switch can be used without a signature, when it is the last command used. In this case the current signature is blanked.

- i <file name and extension>

Sets the input file to use. A directory may be specified. File names may have spaces. File names must have the correct case of letters.

– **w <sample number phase shift number>**

Only included with the professional version.

Sets a change point of phase shift. The first number is a sample during a gap, before the data block with the changed phase shift. The second number is the phase shift, which must be 0 or 180. A sound editor can be used to find a point of change. Unfortunately Gold Wave does not have the capability of showing samples on the X-axis but other shareware programs such as Cool Edit 2000 have this capability.

Samples are numbered from 0. The default phase shift at sample 0 is 0 degrees therefore -w 0 0 has no effect.

My CSW Viewer program can be used to find a quantity of samples up to a point within a gap. Use the range selection boxes with 0 in the first point box and observe the value given for “Time span in samples”.

It is often necessary to use this switch: -w 0 180.

– **s <file number baud rate>**

The value of bits is determined by the quantity of wave variations in a given span of time. This switch sets how the program calibrates its interpretation of the length of that time span.

This switch can be used for cassettes where the baud rate varies between files.

With previous versions of MakeUEF (before V2.3) a calculation on the input wave frequency is done to determine the correct span of samples to expect waves to be near to. E.g. $44100 \text{ frequency} \div 1200 \text{ long waves per second} = 36.75 \text{ samples per long wave}$. The Amateur version still does this. Now an initial read through a CSW file is done to find a mean baud rate.

If you know the mean baud rate of a file, probably from a previous run of MakeUEF, you will be able to specify it. After running MakeUEF with a CSW file you can discover the baud rates within it and then rerun MakeUEF using those baud rates. Probably there is roughly one, baud rate used unless the cassette is a compilation. If the baud rate does not vary much there is no need to do a rerun.

If you find that using an incorrect low baud rate helps you should rerecord with an adjustment to the tone.

– **z <data block number format code>**

Only included with the professional version.

Sets the data format and data block format to use for a block. The first parameter is the number of the data block to use the formats with and all further blocks. The second parameter is the formats to use.

E.g. 5 7E1 would make MakeUEF use a data format of 7 data bits, even parity and 1 stop bit with the fifth and all later blocks. 6 8N2 would make MakeUEF use a data format of 8 data bits, no parity and 2 stop bits with the sixth and all later blocks. The data block format is non-standard with these examples.

BBC is a code for standard BBC/Electron blocks. BBC implicitly has the same data format as 8N1. 8N1 can be used with non-standard block formats to suppress error messages.

ATM is a code for standard Atom blocks. The -y switch will be needed usually to set the baud rate format to 300.

You can read the messages to discover a block number requiring a format change. Trial and error can find the format required. Alternatively you could try looking at the reported security waves to determine the format. The -r switch is useful for seeing if the correct data format has been used.

Supported formats are 7E1, 7E2, 7O1, 7O2, 8E1, 8N1, 8N2, 8O1, ATM and BBC. The blocks are not necessarily the same as audible blocks – the starts and ends of blocks may not be linked to carrier tone but instead to another data block.

Firebird cassettes sometimes have over 40 data format changes. Most cassettes with a non-standard format have only a small amount of changes.

My CSW Viewer program can be used to figure out what data format a block uses.

A block size of 1 is seen as a dummy byte by MakeUEF and does not increment the count of data blocks.

– **r**

Only included with the professional version.

Reports:

- 1: The sizes of data blocks,
- 2: The first pulse and time of the start bit of an unreadable byte.

Pulses are numbered from 0. This option is mainly for doing non-standard blocks and finding the location of waves that are not being read correctly.

– **y <block number baud rate format>**

Only included with the professional version.

The baud rate format can be 1200 or 300 with Acorn cassettes. The baud rate format is 1200 by default.

– **v**

Only included with the professional version.

No verbose messages.

– **u <block number>**

Only included with the professional version.

Turns off detection of a leading carrier tone before a block. This switch is only intended for the version files at the end of Acornsoft Atom cassettes.

It is only possible to have a command from this switch invoked after an Atom format block. It also sets the type of Atom block to be read to data. A message about a bad checksum will probably not appear since MakeUEF ordinarily expects the Atom data block checksum to not be followed by carrier tone. Detection is automatically returned to on for a following block.

– **o <file name>**

Sets the output file to use. A directory may be specified. The extension “hq.uef” will be appended with the professional version and “uef” is appended with the amateur version. File names may have spaces. File names must have the correct case of letters. The file name defaults to the input file name if this switch is not used.

This switch is necessary with sound card input.

– **d**

Only included with the amateur version.

Only for WAV and RAW files.

This option turns on DC offset fixing. This feature positions waves over the zero line if they are away from it, which is common at the start and end of files. The input file has its values fixed and written to a temporary file before the input file is deleted and the temporary file is renamed with the input file name. It is only necessary to fix a files DC offsets once.

The CSW.exe program performs this function better as well as other benefits so it is not recommended to fix DC offsets before using CSW.exe via CSW Viewer.

Do not use this feature with professional cassettes for submission to the preservation project. This switch has been found to be useful with Acornsoft Atom cassettes.

– **f <some number>**

Only included with the amateur version.

If working with RAW sound files, there is no way for the program to find out at what frequency the sound file is recorded. This is necessary information if the frequency is not the default. The number will be assumed to be in decimal unless proceeded with 0x – making it hexadecimal.

The sampling rate is found automatically for other sound file formats and for sound card input.

The default frequency is 44100.

– **t**

Only included with the amateur version.

This is only for RAW files. MakeUEF supports only mono RAW sound files with 8bit samples, but is happy to accept signed or unsigned data. Use of this switch selects unsigned reading of raw data files.

The default data sign type is signed.

– **l**

Only included with the amateur version.

Only for WAV and RAW files and sound card input.

Using this command will read waves using the peak method. The sign reading method is used by default.

The waves will be expected to be at 90 degrees.

– **p**

Only included with the amateur version.

Makes MakeUEF expect inverted waves i.e. 180 degrees. If the -l switch is used in conjunction the waves will be expected to be 270 degrees.

– **h**

Displays the help message, just as passing no command line parameters does. The window displayed can be closed with a key press.

– **b**

Do not beep on errors.

– **c**

Do not compress the UEF file.

HARDWARE REQUIREMENTS

MakeUEF Amateur should work with any sound card that has a Windows driver and supports 8bit data capture at a rate of 44100 or 22050 samples per second. The once common SoundBlaster cards do not have these rates. Adjust your cards mixer via Windows to select the line input socket.

Using a dial-up connection at the same time as using direct sound input may not work. Some computers may require system utilities such as firewalls, anti-virus and anti-spyware to be turned off while using direct sound input. Electron saved files may require more processing power if direct sound input is to be used.

LIMITATIONS

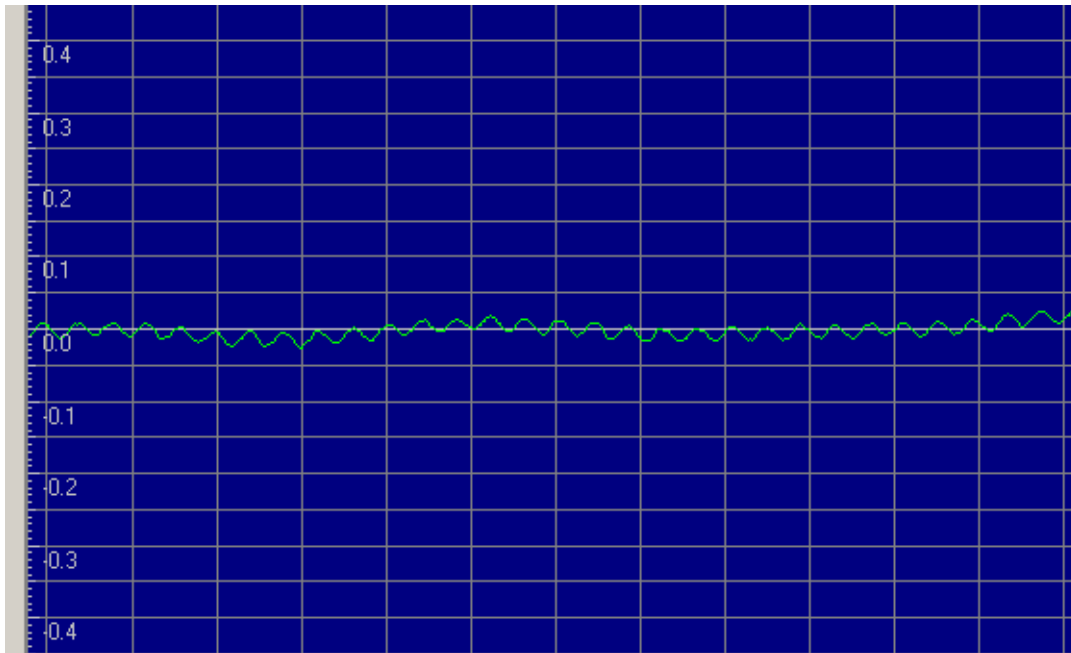
- **Hiss or sound spread from the other side of a tape between files can sometimes be read as security waves or, less likely for hiss, carrier tone. The baud rate for hiss will usually be noticeably different to data blocks. It can be fixed by making the waves unreadable in a gap between data with a sound editor.**

With Gold Wave selecting an area starting at a crossing point of the zero line and then using the filter 'low pass static' with an initial cut-off frequency of 400Hz and a steepness of 5, on the effects menu does this. DON'T entirely silence the area because this does not work well when making CSW files. Alternatively you could rerecord the cassette, playing it with less volume, which should record fewer hisses but that does not work so well with sound spread from the other side of a tape.

With cassettes that have very small gaps you will need to carefully zoom in to the gaps with a sound editor to ensure there isn't hiss. If there isn't any very small gaps, you would only need to filter out the waves if MakeUEF reports that possibly hiss were written, before it finishes.

Many badly recorded cassettes may have a fragmented carrier tone, which IS data to be encoded. The limit of 7 waves for a group prevents hiss and sound spread from the other side of a tape being read most of the time.

Sound spread from the other side of a tape:



- Minor changes in baud rate within data blocks and a following carrier tone are not recorded. Nor are they recorded within leading carrier tones. This is the biggest limitation of the UEF format. CSW files do record minor baud rate changes because they are basically sound recording files.
- **Dropouts on the cassette caused by bad cassette players will NOT make UEF files that fully represent the original cassette. Dropouts are common at the end of the last file of a program, on the trailing tone. If you have a cassette with dropouts only make a UEF file for submission if no one else submits the cassette and also make it known that there are dropouts. Do not be concerned about a fragmented carrier tone unless dropouts caused it.**
- A group of waves must amount to at least 7 or it will be counted as gap.
- The CSW format unifies the amplitude of all waves, so no wave is missed no matter how quiet it originally was. However with other formats used by the amateur version waves with a lower volume than 1/10 of the range will be counted as gap.
- The CSW.exe V2.0 program sometimes produces files with all lowercase names.
- The CSW.exe V2.0 program sometimes produces files with the 'number of pulses' 0.
- Fast sections of carrier tone that are sometimes at the start of a carrier tone are better recorded as a CSW file with a higher than usual high frequency filter. A suitable frequency would be 5000Hz instead of 3000Hz.
- Data blocks are limited to 128K by MakeUEF and by the Acorn cassette block format to 64K.
- Phase shift changing and baud rate changing can only be done at gaps between files. Phase shift changing expects gaps to contain at least a few pulses so that readable pulses are not missed.
- Checksums are not calculated for non-standard blocks whether they have one or not.
- Dummy bytes in carrier tone with a different phase shift to the data cannot be recorded correctly.

- Dummy bytes will only be accepted in the current baud rate format.
- A change of data format cannot be made after a single/dummy byte.
- All cassettes are regarded as having only mono recordings. Some cassettes have only used one of the stereo channels.
- Known cassettes that cannot be made into fully meaningful UEF files include: Welcome, the introductory cassette with the BBC model B which has an error sized wave as the second last wave of the carrier tone before the first data block;

MAKEUEF's DEVELOPMENT AND ITS PROCESSING

Thomas Harte made the original program and hundreds of UEF files have been made with it. Thomas's program appeared to be free from bugs. One mistake was treating all cassettes as inverted sine waves, which made many cassettes difficult to read and this continued for a long time until I thought I noticed a cassette with normal sine waves.

I started rewriting MakeUEF at the beginning of February 2003. I worked on the new version on and off for many months. As I discovered the general format of files the program was made to record it. An excellent discovery was CSW.exe by Ramsoft, which made sound waves more uniform and greatly compacted huge sound sample files. Also it allowed easier reading of the starts of files, which is a particularly bad part to read. CSW.exe had a major upgrade with V2.0, which does compression and records polarity better. The CSW format had to be extended again to hold a creators signature.

The last work done with MakeUEF V1.1 was to create two different versions, one for professional cassettes and one for amateur cassettes. The amateur version is a minimally modified version of the professional version source code using pre-compilation directives and some new code.

After releasing V1.1 it was thought that there are 4 common sine wave types used with Acorn cassettes. I withdrew MakeUEF V1.1 from the public. This again was a mistake. I discovered it is amplifiers with digital IC components that are applying a phase shift to the waves. Fortunately a lot of code I had written was suitable for fixing files recorded with these amplifiers. The next version of MakeUEF professional could have lowered wavelength toleration and have the -p switch removed (which was for 0 degree phase shifts). MakeUEF could then record phase shift positions in the 0115 chunk of the UEF specification, which was changed. A switch similar to the old -p switch, had to be put into MakeUEF V1.2 for inverted sine waves and that is the -w switch.

Support for the non-standard cassettes was written in and MakeUEF had an expanded set of control switches. Electron saved files with occasional extra stop bits was one more peculiarity that had to be catered for. I made a program for viewing CSW files called CSW Viewer and became more aware of the varying baud rates. Towards the end of developing MakeUEF improvements were made to the accuracy of gap lengths and baud rate recording was added. The baud rates are inserted before data blocks and at the start of any other series of readable waves. A switch for calibration by baud rate was one of the last major features added. There was a calibration switch before that used wavelengths in samples.

V1.11 added support for Atom files but improvements weren't made until the V2.x versions. Atom cassettes presented a number of differences to BBC cassettes. All bytes except the last are followed by a short wave. The Atom files therefore have a small difference from a typical KCS file. The block header and checksum has many differences to the BBC's. Acornsoft's 'identification' files do not have a leading carrier tone which is never seen with the BBC. I think this is the Acorn System 1 format.

Theoretically dummy bytes (explained later) might have a different polarity to the data. Also they might have a different baud rate format to the data. Support for these things has not been written in since I haven't seen any cassettes with these things.

The program had to be thoroughly tested with many cassettes. It also had to be carefully examined for errors, which would make UEF files flawed.

I have several UML activity diagrams that have helped with the complex functions. There are various use cases for getting different types of data blocks. That is the programs main function. Sequence diagrams showing different use cases for this have helped with what is the programs most complex part. I also have state diagrams and a specification class diagram for sound classes and CRC cards for all the classes. There is a package diagram for the program and some relationship class diagrams.

Now that MakeUEF is not likely to be extended much I can reflect on some design mistakes.

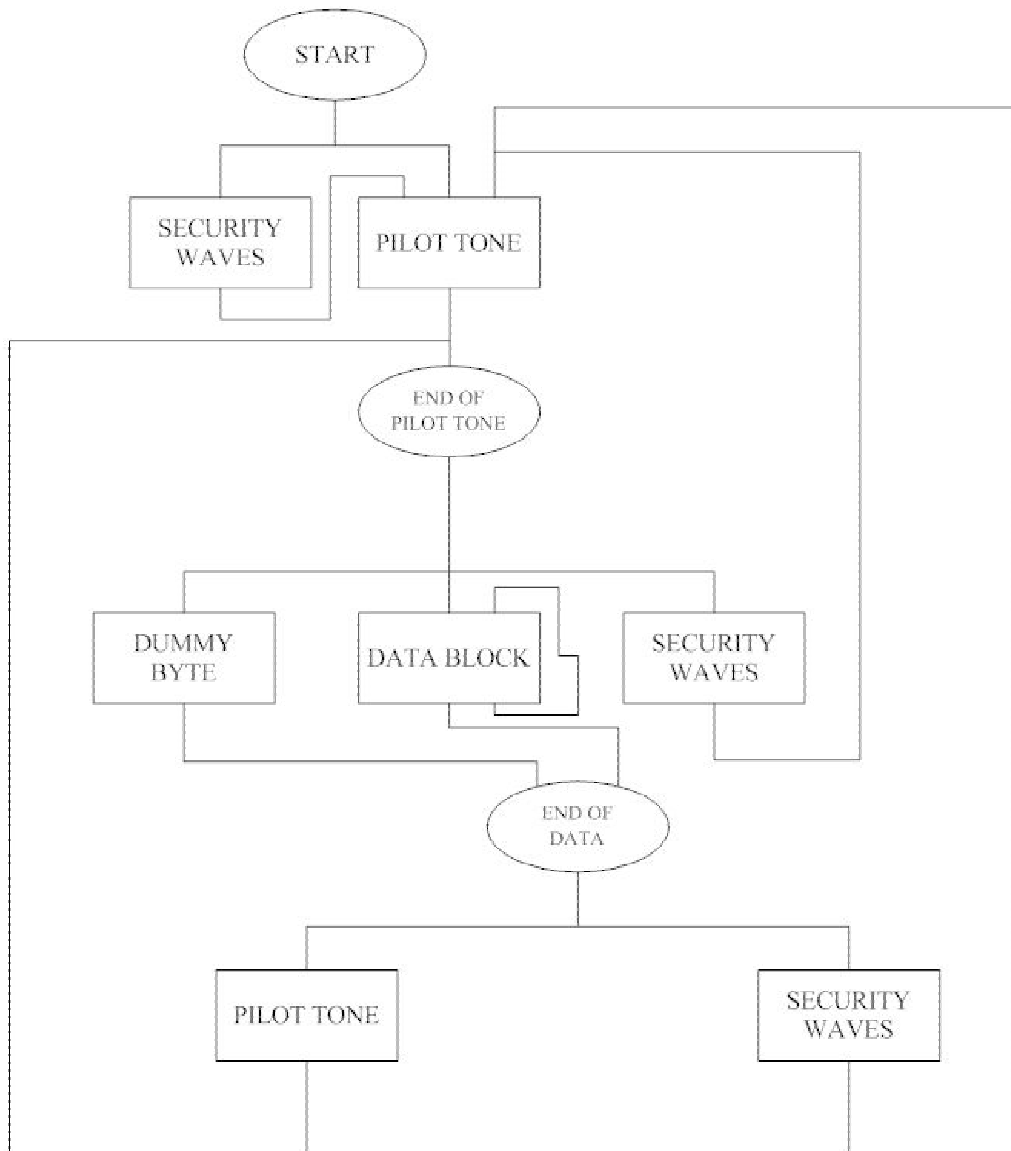
MakeUEF reads from an input file into a small buffer. When the EOF condition is detected the program executes code that deals with what is remaining in the buffer and finishes output files. Although uncommonly used code is separated from the rest, which is useful, this way of dealing with the remaining waves probably has made extra work. It would have been better to use the EOD condition and only finish off the output files after detection.

Treating the standard dummy byte differently to others made the program more complex. There is not a lot gained from using the carrier tone with standard dummy byte chunk of the UEF format.

Most files made with all versions of MakeUEF before V1.9 have some flaws. 300 baud rate format BBC files are ok if made with V1.11 or later. Files with a change of phase shift not at the start of the file are ok if they are made with V1.12 or later.

Following is a flowchart that shows how MakeUEF may recognise blocks to put in a UEF file.

MakeUEFs data recognition flowchart:



In the middle of the carrier tone is almost always, with BBC files, a dummy byte, which fixes a problem with the BBC hardware. BBC OS ROMs before 1.0 do not record a dummy byte. The leading carrier tone is often preceded by a group of watermark waves I call security waves. These waves are sometimes enough to trigger a BBC to detect a carrier tone and read a byte. When this happens a dummy byte is not needed within the carrier tone. Some companies wrote a group of security waves standing alone.

MakeUEF must see a carrier tone with at least 7 waves before it will look for data. Data blocks are normally followed by a carrier tone but security waves will be recorded if there is any.

Gaps will be recorded wherever they are and are omitted to simplify the diagram. After a gap the current position on the diagram is START.

Data blocks comprise of bytes akin to data bytes passed on serial port connections. Ordinary files use the 8N1 data format. One data block can follow another with a different data format.

MakeUEF would record a dummy byte in the current data format, if detected, using chunk &0104. Otherwise a dummy byte in 8N1 format, if detected, would be recorded using chunk &0111. Lastly a dummy byte would be recorded in security waves if the other methods can't be used.

The baud rate is recorded before:

- A carrier tone with dummy byte chunk is written, unless the dummy byte is not followed by carrier tone.
- A data block is written.
- A carrier tone chunk is written except, when it follows a data block, or when the end of the sound recording is near, or when it is followed by a non-standard dummy byte.
- Any group of at least 7 readable waves, possibly starting and finishing with a pulse instead of a wave, followed by a gap or the end of the sound recording.

MakeUEF comprises of about 4000 source code lines in my sound files and other general purpose files and about 7500 source code lines in the MakeUEF project. I have a compact style of source code formatting.

Publications used:

BBC Advanced User Guide P347, P348: Acorn BBC/Electron cassette format.

Wav.doc by Robert Shuler: WAV file format.

Csw.html by Ramsoft: CSW file format.

http://homepage.ntlworld.com/kryten_droid/Acorn/Atom/spa/spa45.htm: Atom cassette formats. Some information is wrong for instance the Atom header flags byte is described as a checksum byte. There is no mention of the carrier tone between the header and the data. This is not online now but can be found via <http://www.archive.org/>.

<http://zlib.net/>: ZLIB compression information.

Wikipedia has some pages of interest. There is one on 'UEF (file format)'. There is another on 'Kansas City Standard'.

COMING IN A LATER VERSION

A GUI.

CHANGE LOG

2.3

- Changed the default expected baud rate with the professional version from 1200 to a value calculated by an initial read of a CSW file.
- Added a message saying that the mean baud rate calculation is being done.
- Added a message reporting the expected baud rate.
- Removed the messages about not using the -s switch.
- Fixed the -p switch in the V2.2 amateur version, which made a crash.
- Fixed direct sound input which did not work in the V2.2 amateur version.
- Fixed missed samples with direct sound input.
- Minor change to the message reporting the expected phase in the amateur version.
- Minor changes to the information given by the -h switch.

2.2

- Fixed a possible but unlikely resource de-allocation problem.
- Changed the printing of non-character ASCII code 127 in file names to a space.
- The checking of the Atom check byte is only done when necessary now.
- Improved dummy byte reporting for Atom Bug-Byte cassettes.
- The help message window can be closed with a key press.

2.1

- Fixed checking of version numbers with DLL files. Only the major number or minor number had to be correct. Had to bump the major version number up to 2.

- Removed accessing of memory in a different module, which is not always allowed by Windows.
 - Fixed a problem with changing from ATM data format to something else.
 - Fixed a problem with finding the length of a file name in a standard Atom block when the block is corrupted.
 - Improved detection of a standard Atom data block check byte near the end of a CSW file not followed by many gap pulses.
 - Changed the reporting of found standard Atom block headers to a similar style as used with standard BBC format blocks.
 - Change the error message about a wrong standard Atom data block checksum to a check byte.
 - Allowed correct reporting of BBC standard file format block numbers between &100 and &FFF. This is seen with 3D Grand Prix.
 - Removed the -x switch but included an error message that reports extra bytes following the data checksum of the BBC standard file format. The block size is always correct as far as I know but the original BBC ROM placed an extra byte at the end of a block.
 - Made corrections to this document for the -g and -u switches.
 - Improved reporting of misread blocks and block numbers. E.g. Postern Pengwyn no screws side.
 - Changed compiler to Borland Turbo Explorer 10.0.2288.42451 Update 2 from C++ Builder 6.
 - Changed the DLL file names for convenience.
 - Removed the program name from the command line in LOG files.
- 1.13
- Fixed a buffer underrun by 1 bug at the EOF point of the input file. This bug was probably only in V1.12 and only had a very small chance of appearing, probably with files without a trailing carrier tone such as Atom files.
 - Fixed missing parts of the message given for the -h command with the Amateur version.
 - Changed the minimum sampling rate to 22050 with the Amateur version.
 - Put the main programs in the DLL files MakeUEF.dll and MakeUEFam.dll.
 - Put the command prompt user interface program in MakeUEF.exe and MakeUEFam.exe.
 - Added better error messages for command errors.
 - Made errors be displayed in a message box instead of at the command prompt window.
 - Made message for the -h switch be displayed on a memo.
- 1.12
- Fixed messages with Electron files with extra stop bits which became not working with V1.11. UEF files produced are not changed.
 - Fixed a gap length bug at a change of phase shift, which was not made at sample 0.
 - Fixed writing of a baud rate of 0 with the Amateur version.
 - Fixed a minor message bug when the signature is blanked sometimes.
 - Removed the -t s command, which was equivalent to the default.
 - Removed the parameter from the -t switch.
 - Improved phase shift changing at the start of files like the improvement with V1.10.
 - Improved the accuracy of wave sizes from WAV and RAW files and sound card input.
 - Changed the REG format code for the -z switch to BBC.
 - Prevented the changing of the baud rate format to the current format, which only produced messages.
 - Prevented the changing of the data format to the current format.
 - Made the program entirely use UML classes and packages.
- 1.11
- Fixed baud rate recording for 300 baud rate format BBC files.
 - Fixed the -d switch, which was not working correctly with the starts of files.
 - Removed writing of baud rate after a 'fill in' '1' bit.
 - Removed writing of baud rate after a standard dummy byte not followed by carrier tone.
 - Changed error messages for the LOG output to have all uppercase letters.
 - Removed the -l s command, which was equivalent to the default.
 - Removed the parameter from the -l switch.
 - Made a big improvement to the speed of the -d switch.
 - Added support for Atom files. 1200 baud rate format Atom files have not been tested.

- Added support for leaderless version files on Acornsoft Atom cassettes with the new -u switch.
 - Made the program mostly use UML classes and packages.
- 1.10
 - Fixed messages with Electron files with extra stop bits which became not working with V1.9. UEF files produced are not changed.
 - Fixed the -d switch, which became not working with V1.9.
 - Fixed detection of start bit from 2nd byte of a data block onwards. This was accepted without correct size checking.
 - Improved detection of a byte near the end of a file not followed by many gap pulses.
 - Improved phase shift changing by inserting a dummy 0 length pulse at the start of the buffer instead of accepting the front pulse as a gap pulse without looking at it.
 - Changed the -r switch to report the first pulse and time of an unreadable byte.
 - Made a wider range of data be used with the arithmetic for the -d switch.
 - Compressed EXE files with UPX.
- 1.9
 - Fixed a minor bug with parsing the -s switch.
 - Fixed a bug with the sequence: carrier tone, non-standard dummy byte, 2 carrier tone waves then gap. This is only an uncommon sequence, which might come from a rough carrier tone.
 - Fixed some uncommon sequences of readable waves.
 - Fixed a bug in certain sequences with recording a carrier tone of the minimum group size which was recorded as 1 too many.
 - Prevented use of 0 with the -s, -f, -y and -z switches.
 - Improved the accuracy of the reported time while less than 100 seconds with the -r switch.
 - Improved interruption checking which included adding a message to UEF files.
 - Improved standard data block checking.
 - Added detection and recording of a final pulse in certain sequences.
 - Added the -x switch.
- 1.8
 - Fixed bugs with uncommon wave sequences following a data block.
 - Fixed a bug with the reporting of the pulse number of a last byte with the -r switch.
 - Fixed a buffer overrun bug with non-standard byte analysing at the end of the sound recording. This would not realistically have been encountered.
 - Fixed a bug with 7 bit data files.
 - Fixed writing of leading carrier tone and dummy byte with MakeUEFam from 2nd file.
 - Fixed bug with input files with only 0, 1 or 2 pulses.
 - Removed excessive writing of the baud rate with non-standard dummy bytes.
 - Changed precision of floating point values in the LOG file to 7 from 5 (6 for initial gap).
 - Minor changes to output messages.
 - Added support for 300 baud rate format BBC files.
 - Added reporting of the time of the pulse number of a last byte with the -r switch.
 - Added bounds checking on data block storage area.
 - Linked with zlib V1.2.3.
- 1.7
 - Fixed a bug with CSW files with no data.
 - Fixed a bug with security waves near the end of the sound recording.
 - Fixed a bug with changing the phase shift at the end of the sound recording.
 - Fixed a buffer underrun bug while analysing a byte at the end of the sound recording.
 - Fixed a gap length bug while finding a carrier tone at the end of the sound recording.
 - Fixed a gap length bug while processing security waves straight after a data block at the end of the sound recording.
 - Added the MakeUEF program name to the top of LOG files.
 - Added a beep upon major errors.
- 1.6
 - Fixed a bug with baud rate writing. This was a STL library bug.

- Added writing the date and time to LOG files.
- 1.5
 - Fixed a bug with uncompressed CSW files.
 - Added writing the command line used to the top of LOG files.
 - Added writing the creator's signature to the top of LOG files.
- 1.4
 - Fixed a bug with an error wave following a data block.
 - Fixed a bug with an error wave following a dummy byte.
- 1.3
 - Removed reporting of security waves type (pulse or wave) where it is inappropriate.
 - Changed the reporting of the number of pulses, to report a start bits first pulse, of a currently unreadable byte.
 - Added recording of the length of the gap at the start of a cassette and the trailing gap after the last readable waves.
 - Changed recording of a leading carrier tone with MakeUEFam to record a dummy byte.
 - Changed type of gap block recorded by MakeUEFam to floating point.
 - Changed the UEF minor version to 10 with MakeUEFam.
 - Fixed a bug with CSW files and MakeUEFam.
 - Fixed a baud rate bug.
 - Fixed a bug with security waves following a dummy byte.
 - Fixed minor inaccuracy bug with wave calibration.
 - Fixed a bug with carrier tone detecting.
 - Made dummy bytes get recorded in 8N1 format if possible when the current format cannot accept the byte. If neither data formats are suitable security waves are used.
 - Changed compiler to C++ Builder 6 from 4.
- 1.2
 - Removed the -p switch from the professional version.
 - Removed creation of temporary uncompressed UEF file.
 - Changed the default sine wave phase shift to 0 degrees. It was formerly +180 degrees.
 - Changed chunk &0115 in the UEF format specifications to record phase shift.
 - Reduced wavelength toleration to +/- 25% for short waves and +/- 15.38% for long waves from +/- 28.57% for all waves, with the professional version.
 - Added support for CSW files with signatures.
 - Added reporting of gap lengths written.
 - Added reporting of carrier tone lengths written.
 - Added output to LOG file as well as the screen.
 - Fixed a bug with a long wave of security waves following a data block, which is never actually seen.
 - Some improvements to the readability of the output log.
 - Fixed the -lp switch and peak-to-peak reading, which did not work.
 - Fixed a minor bug with sound card input.
 - Added the -w switch.
- 1.1
 - No functional changes.
 - Rewrote the file handling code.
 - Fixed bugs with DC offset fixing at the ends of files.
- 1.0:
 - Extensive rewriting by a second programmer.
 - Now MakeUEF has a professional cassette version and an amateur cassette version.
 - Now MakeUEF accurately records the data.
 - Now MakeUEF accurately records the gap lengths.
 - Now MakeUEF accurately records the mean baud rates.
 - MakeUEF is now written in C++. The sound card input code is mostly unaltered from the previous version.
 - Files with changing names are now supported.

- Files with occasional extra stop bits between bytes saved by an Electron are now supported and messages are shown. Files with occasional extra stop bits between bytes that are different to the typical ones produced by the Electron may not have a message shown but they are rare. E.g. Electron Pengwyn.
- Files with negated polarity are now better supported.
- Less wavelength toleration, which forces the use of the correct polarity and better sound files.
- Out of sequence block numbers are now supported.
- Incorrect 'last block flag' blocks are now supported.
- Changes of data format within a block are now supported. E.g. Firebird cassettes and The Lost Crystal.
- Signatures can now be added to UEF files.
- Changes to the calibration can now be made between files.
- A new block was added to the UEF format to support bundles of long and short waves with or without starting or finishing with a pulse.
- A new block was added to the UEF format to support files with non-standard data formats.
- A new block was added to the UEF format to support changes of polarity.
- A new block was added to the UEF format to support more accurate gap lengths.
- A change was made to the UEF format to the baud rate block for greater accuracy.
- Support for CSW files was added.
- Messages about the input file format are now recorded in UEF files.
- More extensive checking of input file headers and various error messages was added.
- The switches -c, -d, -g, -p, -r, -v and -z are new.
- Changes to defaults.
- Removed carrier tone and sample length calibration and added baud rate calibration with the -s switch.
- Added a message recorded in UEF files if the -s switch is not used and a message on the screen.
- Added a hiss written message that appears on the screen.
- Added a 'possible dropout' message that appears on the screen for short gaps.

0.3b:

- small fixes to Windows code, hopefully will no longer crash on exit

0.3:

- iterated towards fixing windows version. Does it work for you?
- added alternate 'peaks' frequency measure for greater reliability on old cassettes
- added alternate 'tones' encoding, someone may want it
- allowances for surprise non-ending of files (Elite now closer to working)
- fixed cassette image encoding so that high tone hangs for a bit after files, and gaps lie between files - as with real cassettes
- adding filename checking for illegal characters before printing names to the VDU - in particular trying to print the name of the '?' file on Acornsoft titles was causing the on screen prompts not to work

0.2:

- completely rewrote data parsing algorithm to sync properly with cassette
- altered sound card code to select > 22050Hz frequencies where possible in DOS & Windows

0.1:

- first release

CONTACTING THE PROGRAMMERS

Don't send any large WAV files but FLAC files will be fine. CSW files are ok but are not good for assessing the phase shift used or the suitability of the tone and volume.

You may need to enable breaking up of large emails. With Outlook Express it is an option for a mail account on the advanced tab. I prefer to receive file attachments with a separate email to messages so I can delete emails with an attachment and keep the messages.

If you have seen what looks like a bug send:

- 1: the input file,
- 2: the command line used,
- 3: the LOG file.

Designer and programmer:

Fraser Ross, mail to: [fraserross@f2s.com?subject=MakeUEF V2.3](mailto:fraserross@f2s.com?subject=MakeUEF%20V2.3)