Extracting data from .dtg format files

Mark Johnson
21 July 2008

DTG files are packed binary archive files containing tag sensor and audio data. Each file contains a number of timing and data consistency checks, and so successful extraction of the data from a DTG file is a good indication of data quality. Each tag deployment may produce up to 24 DTG files which will have names of the form *sw253a01.dtg*, *sw253a02.dtg*, for deployment sw253a.

Two versions of the DTAG have been produced, Version 1 and Version 2. Each produces distinct format DTG files and the correct extraction software must be selected. Version 1 DTAGs have tag ID numbers from 1 to 20 and require the *tagread2b.exe* or the *tagrd32.exe* program for data extraction. Version 2 tags have 3 digit ID numbers with 2 as the first digit. Version 2 DTG files are unpacked using *ffsrdall.exe*. Both executables should run on any Windows PC following the procedures described below. Please report any misbehaviour of these tools to *markjohnson@st-andrews.ac.uk.*

Unpacking Version 1 DTG files

Always try extracting with *tagrd32.exe* first. If *tagrd32.exe* displays an error, try *tagread2b.exe.* This is only necessary for some of the first generation tag deployments. Follow the same steps as *tagrd32.exe.*

1. Check the size of the DTG file and note the corresponding number of blocks in the file according to the following table:

Approx. File Size	NBLOCKS
136 MB	8192
68 MB	4096
34 MB	2048
17 MB	1024

If the file size is between two of the sizes shown, choose the NBLOCKS corresponding to the next larger file size.
2. Copy *tagrd32.exe* to the directory containing the DTG file to be unpacked. Make sure this is a directory that can be written to, e.g., that it is not a read-only CD.
3. Double-click on *tagrd32.exe* and enter the full name (including .dtg extension) of the file to be unpacked. Type return.
4. Hit return at the next prompt to select the default output file name.
5. Type in the sampling-rate in kHz (with no decimal point) at the following prompt followed by return. Consult the DTAG experiments list for the sampling-rate used in each deployment.
6. Type in the value of NBLOCKS from the above table at the final prompt followed by return.

7. The screen will show some information during the extraction. Wait until the screen shows:

```
Successfully terminated
Press any key to close...
```

This indicates that the unpacking procedure completed successfully and that the extracted data is ready to read. Two output files should have been produced in the same directory as the input file and with the same name but with .SEN and .WAV suffixes. The .SEN file contains sensor data in a packed binary format. The .WAV file is a standard WAV-audio format file containing the audio recording. The .SEN file can be read in Matlab using the tag toolbox function *readsen.m*. A calling example is:

s = readsen('/tag/data/sw253a13.sen') ;

Type *help readsen* for other calling options and for a description of the output matrix, s. The .WAV file can be read by most audio editing packages (Cool Edit, Gold Edit etc.). Sections can be read into Matlab using the wavread16.m tag toolbox function. A calling example is:

x = wavread16('/tag/data/sw253a13.wav',fs[st ed]) ;*

where fs is the sampling rate of the audio data (e.g., 32000) and st and ed are the start second and end second that you wish to read. Be aware that audio data takes considerable memory and that it is not generally practical to read more than 1 minute or so of data at a time. Type *help wavread16* for other calling options and for a description of the output matrix, x.

Unpacking Version 2 DTG files

1. Copy *ffsrdall.exe* to the directory containing the DTG file to be unpacked. Make sure this is a directory that can be written to, e.g., that it is not a read-only CD.
2. Double-click on *ffsrdall.exe* and enter the filename base (eg. sw123a) of the file to be unpacked. Type return.
3. The prompt will ask you which chips to convert. Type A and return to convert all of the chips from that deployments.
4. Hit return at the next prompt to select the default output file name.
5. Hit return at the next prompt to extract all data.
6. Wait until extraction is complete.
7. The screen will show some information during the extraction. Wait until the screen shows:

```
Processed mmmm of nnnn blocks
Press any key to close...
```

where mmmm and nnnn are usually similar 4 digit numbers. This indicates that the unpacking procedure completed successfully and that the extracted data is ready to read. Three output files should have been produced in the same directory as the input file and with the same name but with .SWV, .WAV and .TXT suffixes. The .SWV file contains multi-channel sensor data in a WAV-audio format. The .WAV file is a standard WAV-audio format file containing the audio recording. The .TXT file is a log containing timing information and error status of each chunk of data unpacked from the DTG file. See *dtagfilesystem.doc* for a description of the fields in this file. The log file is normally only inspected during initial data quality assesment and during formation of the cue table (see *tag2whale.doc* for details of this process). An overall log file will also be generated with a name of the form ***tagid****log.txt*, where ***tagid*** is the base name of the deployment that you entered at step 2. This log file contains a report of any errors found when extracting the data and should be checked through. It also contains the first and last audio sample value in each chip which should be checked: the last samples of chip n should always be the same as the first samples of chip n+1 (there is a deliberate 1 sample overlap between chips to allow error checking).

The .SWV files can be read using most audio editing packages (Audacity, Cool Edit, Gold Edit etc.) or in Matlab using the tag toolbox function *swvread.m*. A calling example is:

```
s = swvread('/tag/data/sw160a07.swv') ;
```

Type *help swvread* for other calling options and for a description of the output matrix, s. The .WAV file can be read by most audio editing packages. Sections can be read into Matlab using the wavread16.m tag toolbox function. A calling example is:

```
x = wavread16('/tag/data/sw160a07.wav',fs*[st ed]) ;
```

where fs is the sampling rate of the audio data (e.g., 96000) and st and ed are the start second and end second that you wish to read. Be aware that audio data takes considerable memory and that it is not generally practical to read more than 30 seconds or so of data at a time. Type *help wavread16* for other calling options and for a description of the output matrix, x.

Executables required

tagread2b.exe or tagrd32.exe (Version 1 DTAGs)
ffsrdall.exe (Version 2 DTAGs)