

## Header Utility

The GRAVityFCAVTest and MatrixA818Test programs have a Header Utility that replaces the "Load Header" and "Get Header" functions used in older versions of these programs. This utility allows the user to open a header data file, edit the data, save the data, load the data into a card, and read data from the card. The Header Utility can be used to create new header data files, as well as variations on existing data files. The GRAVityFCAVTest and MatrixA818Test programs, as well as some of the example programs supplied with the Gravity and Matrix SDKs, use a "Header Format Definition" file to define the display format of header data. The Header Utility makes use of Header Format Definition (HFD) files, which complement the header data files. This document explains the usage of the Header Utility as well as the format and content of HFD files.

### Using the Header Utility in **GRAVityFCAVTest** and **MatrixA818Test**

When the Header Utility is executed by pressing the "Header" button on the main form, the program will attempt to automatically find an appropriate HFD file and header data file. The program will first look for a HFD file with the same name as the Video Mode (much like the "Auto Load Header" feature of the program looks for a data file). If that fails, it will look for a file named with the "base" portion of the Video Mode. For example, for mode ASVI\_GTF3, the Header Utility will first look for a HFD file named "ASVI\_GTF3.hfd", then look for a file named "ASVI.hfd". If neither of those are found, the Header Utility will use the HFD file named "default.hfd" if it exists, or load default header format values if not. The Header Utility follows the same logic when looking for a header data file to load. In the case of the ASVI\_GTF3 mode, it will first look for "ASVI\_GTF3.txt", then "ASVI.txt", then "default.txt". If none of these are found, the header data will be all zeros (0).

### How to use the Header Utility to load a header into the card

Using the Header Utility to Load (write) the header to the card:

- Press the "Header" button. The program will load header data for the current Video Mode, and display the data using the correct format.
- If the current data is acceptable, simply press the "Write" button. The checkboxes next to the "Write" control which buffers will get the header. Normally headers are written to both Ping and Pong, so this is the default setting for the check boxes.
- You may also alter the current data before you load it, if desired. Simply perform the desired edits, and then press the "Write" button.

### **How to use the Header Utility to get header data from the card**

- Press the "Header" button. The program will load header data for the current Video Mode, and display the data using the appropriate format.
- Select the buffer to read using the radio buttons next to the "Read" button.
- Press the "Read" button.
- The data will be read from the card and displayed. If you wish to save this data, press the "Save" button.

### **How to use the Header Utility to edit an existing header data file**

- Press the "Header button. The program will load header data for the current Video Mode, and display the data using the correct format for this mode.
- If you wish to edit a different data file, press File - Open and select the desired file.
- Make the desired changes, then press the File - Save button. The program will prompt you for confirmation because you are overwriting an existing data file.

### **How to use the Header Utility to create a new header data file**

- Press the "Header button. The program will load header data for the current Video Mode, and display the data using the correct format for this mode.
- If you wish to create your new data file using a different existing data file, press File - Open and select the desired file.
- Make the desired changes, then press the File - Save button. Enter the file name for the new file. The program will prompt you for confirmation if you use an existing file name. Take care not to unintentionally overwrite the standard data files for the video modes, unless that is your intention.

### **How to use the Header Utility to display header data using alternate formats**

The Header Utility will use the correct format for the current video mode. If you wish to display the data using another format, press the Format – Open button and select the format file you wish to use to display the data.

### **How to use the Header Utility to display alternate header data**

The Header Utility will load the data file for the current mode. If you wish to display alternate data, press the File - Open button and select the data file you wish to display.

## The Header Format Definition (HFD) file

The Header Format Definition (HFD) file defines the format of the header data. This format is used to display the data for ease of editing by the user. The header format is totally independent from the raw header data itself, except that the header data file loaded into the card should contain at the same number of defined data words in the format used. If the header contains less than the number of words defined in the format, the data will be padded to the length defined in the format. If the header data file contains more than the number of words defined in the HFD file, the excess words will be ignored. HFD files are named using the "hfd" extension (i.e. "FCAV14.hfd" or "AIRINC818.hfd").

It is important to remember that the Video Mode and Header Format Definition file determine the size of the header, not the number of words in the header data file. Also, the HFD file for each mode can be tailored for the specific video mode if desired. If it is preferable to have video modes "share" a common HFD file, all that is required is to create a single HFD file using the "base" name of these video modes. For example, all of the ASVI modes (i.e. ASVI\_DMT, ASVI\_GFT1, etc.) can share the "ASVI.hfd" file.

### Example Header Format Definition file

The contents of a typical HFD (Header Format Definition) file are shown below. This example file contains comments explaining some of the contents and elements of the file.

```
.....  
,  
"  
'" Header Format Definition file  
,  
' Comments begin with the "'" (single quote) char. All other lines are assumed to be  
' format definitions.  
,  
' Types of records in this file:  
' 'f', format identifier  
' 'n', number of words in header  
' 's', section definition  
' 'w', Word definition  
' 'u', user area definition  
' 'x', extended data definition (contents TBD)  
,  
"  
'" The 'f' record  
' The 'f' record contains the format identifier string:  
' 'f', format identifier  
,  
"  
'" The 'n' record  
' The 'n' record contains the number of words in the header:  
' 'n', number of words in header
```

```

'
''' The 's' record
' Each 's' record contains a section identifier and number of words in section:
' 's', section identifier, number of words in section ...
'''
'
''' The 'u' record
' Each 'u' record contains the parameters for an "undefined" area. An undefined
' area is specified as part of the header but has no word definitions. This
' record contains the word offset, the number of words, and the identifier of
' the undefined area as follows:
' 'u', word offset, number of words, undefined area identifier
'''
'
''' The 'w' record
' Each 'w' record contains the word index and word name:
' 'w', Word offset, word identifier, ...
' followed by any number if bit field definitions:
' ..., field offset, field length, field name, ...
'
' Note that field offset and field length in the word definition are in bits.
' The 3 field parameters repeat for the number of bit fields in the word.
'''
'
''' The 'x' Definition
' The 'x' record contains information about the extended header data. This
' record is included for future use and it's contents of this have not yet
' been determined. It is intended to be used for extended header data which
' may include color palette or other information.
'''
'
''' This file currently only defines the first 32 words (128 bytes) of the
' header. The Word Identifier and Field Identifier may be left blank. The
' order or appearance of the field definitions on screen can be controlled
' by the order they are listed in the word definition.
'
.....
' format identifier:
f, Default Header Format
'
' number of words in header:
n, 532
'
' FC Frame Header section (6 words, 0 through 5)
s, FC Frame Header
w, 0, FC Frame Header (word 0), 8, 24, D_ID, 0, 8, R_CTL,

```

w, 1, FC Frame Header (word 1), 8, 24, S\_ID, 0, 8, CS\_CTL,  
w, 2, FC Frame Header (word 2), 8, 24, F\_CTL, 0, 8, TYPE,  
w, 3, FC Frame Header (word 3), 16, 16, SEQ\_CNT, 8, 8, DF\_CTL, 0, 8, SEQ\_ID,  
w, 4, FC Frame Header (word 4), 16, 16, RX\_ID, 0, 16, OX\_ID,  
w, 5, FC Frame Header (word 5), 0, 32, Parameter

' Object 0 Header section (22 words, 6 through 27)

s, Object 0 Container Header

w, 6, Object 0 Container Header (word 0), 0, 32,  
w, 7, Object 0 Container Header (word 1), 0, 32,  
w, 8, Object 0 Container Header (word 2), 0, 32,  
w, 9, Object 0 Container Header (word 3), 0, 32,  
w, 10, Object 0 Container Header (word 4), 0, 32,  
w, 11, Object 0 Container Header (word 5), 0, 32,  
w, 12, Object 0 Container Header (word 6), 0, 32,  
w, 13, Object 0 Container Header (word 7), 0, 32,  
w, 14, Object 0 Container Header (word 8), 0, 32,  
w, 15, Object 0 Container Header (word 9), 0, 32,  
w, 16, Object 0 Container Header (word 10), 0, 32,  
w, 17, Object 0 Container Header (word 11), 0, 32,  
w, 18, Object 0 Container Header (word 12), 0, 32,  
w, 19, Object 0 Container Header (word 13), 0, 32,  
w, 20, Object 0 Container Header (word 14), 0, 32,  
w, 21, Object 0 Container Header (word 15), 0, 32,  
w, 22, Object 0 Container Header (word 16), 0, 32,  
w, 23, Object 0 Container Header (word 17), 0, 32,  
w, 24, Object 0 Container Header (word 18), 0, 32,  
w, 25, Object 0 Container Header (word 19), 0, 32,  
w, 26, Object 0 Container Header (word 20), 0, 32,  
w, 27, Object 0 Container Header (word 21), 0, 32,

' Object 0 Ancillary Data section (4 words, 28 through 31)

' Ancillary data

s, Object 0 Ancillary Data

w, 28, Object 0 Ancillary Data (word 1), 0, 32,  
w, 29, Object 0 Ancillary Data (word 1), 0, 32,  
w, 30, Object 0 Ancillary Data (word 2), 0, 32,  
w, 31, Object 0 Ancillary Data (word 3), 0, 32,

' Undefined area

u, 32, 500, Undefined Area

' Extended data (TBD) section (??? words, 32 through ???)

x, Extended Data section

.....