2 C3D DATA FILE FORMAT

2.1 FILE STRUCTURE

The *SID* program generates files having the extension .C3D that contain the 3D data computed by the AMASS software. This chapter, together with Chapter 3 which describes the subroutines available for reading and writing data in the ADTECH parameter format, provides the necessary information to allow the user to write programs to access the data.

The coordinate and analog data in .C3D files are written in 16-bit integer (INTEGER*2) format, or optionally floating point (REAL*4) format, consisting of records that are 512 bytes long. The files have three sections. The first section consists of just one header record and contains a few parameters as described in Section 2.2. The second section starts at record number 2 and contains all of the parameters obtained from the .DES parameter file as well as others added by the *SID* program. This section is variable in length but is typically about 10 records long. The rest of the file contains the 3D point coordinate data, and analog data if analog data were available to the *SID* program when it created the .C3D file. Normal output of data from the *SID* program is in the integer format. However, the facility for the data to be written in REAL*4 format is also available for the storing of filtered data where the integer form may not provide sufficient precision.

2.2 HEADER RECORD

Although all parameters of interest are contained in the parameter section, a few essential parameters necessary for accessing the data in the file are repeated in the header record in simple form. The parameters in this record are useful if one does not wish to use the parameter subroutines required to access the quantities written in parameter format. The ADTECH parameter format allows for very convenient interactive user access, but is more complex from the programming aspect. The first few words of the header record are utilized to store the abbreviated parameter data, and the rest of the header record contains zeros unless time event data have been written to the data file by the *ADG* program. The parameter values stored in the header record are as listed in Table 2.1. The 16-bit words hold integer values unless otherwise noted.

WORD	CONTENTS
1	Byte 1: number of 1st parameter record Byte 2: must contain $80_{decimal}$
2	Number of 3D points
3	Number of analog channels / **video** frame
4	First frame of .SEG file transferred to .C3D file
5	Last frame of .SEG file transferred to .C3D file
6	Maximum interpolation gap (frames)
7,8	(REAL*4) Scale factor for converting integer 3D data to reference system units. If negative, 3D data is already in REAL*4 format.
9	Starting record number for 3D point and analog data
10	Number of analog frames / video frame
11,12	(REAL*4) Video frame rate (Hz)
13-149	Currently not used
150	$12345_{decimal}$ (keyword to identify presence of event data)
151	Number of defined time events -- maximum of 18
152	Not used
153-188	(REAL*4) Event times (in seconds) -- maximum of 18
189-198	(BYTE*1) Event display switches (0=ON, 1=OFF)
199-234	Event labels. 4 bytes for each event.

Table 2.1 -- Contents of first record of .C3D file.

The number, times of occurrence, and labels of any time events defined during the use of the graphics display program *ADG* are stored in the first record if *ADG* is exited by use of the **`EX`** command. Time events are described in Chapter 6 of the ADG User's Manual. If time events are stored in this record, their data are stored as described in Table 2.1.

2.3 PARAMETER RECORDS

The parameter section is written in ADTECH parameter format which allows the user to interactively access the parameters with the program *PRM*. Programmers may access the parameters through the FORTRAN library PRMLIB distributed with AMASS, or write programs to access the parameter data directly. Chapter 3 provides instructions for calling the subroutines in the PRMLIB library, and the rest of this section describes in detail the ADTECH parameter format for those that want to access the parameters the hard way.

The first two bytes of the first parameter record are either not used, or form the first two bytes of the file. If they are the first two bytes of the file (not the case for C3D files), then **byte 1** contains the number of the first parameter record, and **byte 2** contains 80_{decimal}.

Byte 3 of the first parameter record contains the number of parameter records, and **byte 4** contains 85_{decimal} + `processor_type`. `Processor_type` may have the value 1, 2, or 3, and is defined in Section 3.1. The actual parameter data start at byte 5 of the first parameter record.

Within the parameter records, the data belonging to groups and parameters are stored in no particular order, and are located by searching through the total data block. The storage unit is the byte.

Group data are stored in the format shown in Table 2.2:

START BYTE	LENGTH (Bytes)	DESCRIPTION
1	1	Number of characters in group name
2	1	Group ID number (negative)
3	n	Group name
3 + n	2	(2-byte integer) offset in bytes pointing to start of next group/parameter
3 + n + 2	1	Number of characters in group description
3 + n + 3	m	Group description

Table 2.2 -- Storage format for group data.
n represents the number of characters in group name.
m represents the number of characters in group description.

Parameter data are stored in the format shown in Table 2.3:

START BYTE	**LENGTH (Bytes)**	**DESCRIPTION**
1	1	Number of characters in parameter name
2	1	Group number (positive) to which parameter belongs
3	`n`	Parameter name
3 + *n*	2	(2-byte integer) offset in bytes pointing to start of next group/parameter
3 + *n* + 2	1	Length in bytes of each data element, = -1 for **C**haracter 1 for **B**yte 2 for **I**nteger*2 4 for **R**eal*4
3 + *n* + 3	1	Number of dimensions of the parameter (maximum = 7) = 0 if the parameter is undimensioned.
3 + *n* + 4	`d`	Parameter dimensions
3 + *n* + 4 + *d*	`t`	The parameter data.
3 + *n* + 4 + *d* + *t*	1	Number of characters in description
3 + *n* + 4 + *d* + *t* + 1	`m`	Parameter description

Table 2.3 -- Storage format for parameter data.
n represents the number of characters in group name.
m represents the number of characters in group description.
d represents the number of dimensions of the parameter ($0 \leq d \leq 7$).
t = product of all dimensions and element length

The *PRM* program responds to the command **`SH`**, which will dump all parameter records in byte format, and their ASCII equivalent, to the screen.

2.4 DATA RECORDS

The 3D coordinate and analog data are written frame-sequentially starting at the beginning of the first data record. The data are packed into records such that frames **may** cross record boundaries. The 3D coordinate data are normally stored in 16-bit integer format and must be multiplied by the POINT:SCALE factor to generate values expressed in the external (reference coordinate system) measurement units. If the data is stored in floating point format (REAL*4), then the scale factor has already been applied and is it set to be negative. The *ADG* program will automatically interpret the data, integer of floating point, as required. The analog data are not processed by either the *SEG* or *SID* programs and are stored in the .C3D file in raw form as output by the data capture software. Scaling and offsets for the analog data may be specified in the ANALOG: group in the .DES parameter file. *ADG* applies these parameters when the data are accessed.

Each 3D point is described by four words. If the data is stored in integer form, these 16-bit words contain the following:

WORD	CONTENTS
1	X-coordinate of point ÷ scale factor
2	Y-coordinate of point ÷ scale factor
3	Z-coordinate of point ÷ scale factor
4	Byte 1: cameras that measured marker (1 bit for each camera) Byte 2: average residual for point measurement ÷ scale factor

Table 2.4 -- 3D point data contents

NOTES:

- In Byte 1 of Word 4, bits are set corresponding to which cameras contributed to the point's measurement. Bit 0 refers to the first camera, bit 1 the second, etc. (Example: if cameras 1, 2, 3 and 5 contributed to a point's measurement, Byte 1 of Word 4 would be set to $0010111_{\text{binary}} = 23_{\text{decimal}}$.)

- If a point was invalid in a frame (not observed by at least two cameras), its 4th word will be set to -1, and the X, Y, and Z coordinates will be set to zero. For points that are interpolated by the *SID* program, the residual is set identically to zero.

- In an integer file the coordinates and residuals are recorded in internal units, and must be multiplied by the scaling factor in POINT:SCALE to obtain reference coordinate system units. Within each video data frame the points are stored in their proper order, as assigned in *SID*, followed by the analog data (ordered by frames) if any were present.

- The analog data are stored in the same order as in the raw data file, i.e., if the analog frame rate was higher than the video frame rate, the order through the channels is repeated as many times as is appropriate. The first word after the end of the analog data is the X-coordinate of the first point in the next frame.

- The utility program RDFRME described in Section may be used to list the point coordinates and analog channels in each frame.

- If the data is stored in floating point format, the X, Y, and Z coordinates have already been multiplied by the scale factor. The 4th word is the normal 4th word integer value stored as a floating point number. To extract its data, it should be converted to integer, divided into high and low bytes, and the low byte must be multiplied by the scale factor to obtain the correct residual value.