Data File Standard for Flow Cytometry, Version FCS3.0

Data File Standards Committee of the International Society for Analytical Cytology (ISAC)

FCS version 2.0 can be found in Cytometry 1990;11(3):323-32.

ABSTRACT

The flow cytometry data file standard provides the specifications needed to completely describe flow cytometry data sets within the confines of the file containing the experimental data. In 1984 the first Flow Cytometry Standard format for data files was adopted as FCS1.0, this standard was modified in 1990 as FCS2.0. We report here on the proposed next generation Flow Cytometry Standard data file format, FCS3.0. The principal goal of the Standard is to provide a uniform file format allowing files created by one type of acquisition hardware and software to be analyzed by another type. The proposed FCS3.0 standard maintains backwards compatibility with previous versions by retaining the basic FCS file structure. The FCS structure requires that each data set in a file contains three segments: HEADER, TEXT and DATA, with an optional ANALYSIS segment. The HEADER appears first and contains plain text byte offsets needed to locate the other segments. The TEXT segment contains plain text keyword-value pairs that describe the experiment, the instrument, the specimen, the data and any other information which the file creator wishes to include. The DATA segment contains the actual FCM data in one of several formats specified in the TEXT segment. The ANALYSIS segment contains plain text keyword-value pairs that describe user-specified analyses of the data. The proposed changes in FCS3.0 include: a mechanism for handling data sets of 100 megabytes and larger, support for UNICODE text for keyword values, support for cyclic redundancy check (CRC) validation for each data set, a requirement for the inclusion of information describing the method of signal amplification and increased support for the inclusion of time as a measurement parameter.

INTRODUCTION

The goal of the Flow Cytometry Data File Standard is to facilitate the development of software for reading and writing flow cytometry data files in a standardized format. Application of a standard file format allows files created on one type of instrument to be read and analyzed by software implemented on a different computer. The original FCS standard was published in 1984 as FCS1.0 (1) and amended in 1990 as FCS2.0 (2).

The changes included in FCS3.0 were made necessary by rapid evolution in microcomputer technology, computer communications, instrument design and experimental complexity. These technological advances have resulted in an increase in the average data file size straining the limit of 99,999,999 bytes per data set designed into previous FCS versions. FCS3.0 provides a mechanism to avoid this restriction while retaining backwards compatibility for those data files which do not exceed the 100-megabyte limit. The growth of computer networks has resulted in the routine movements of large amounts of data between computers. This has created the need for a means of confirming file integrity. Therefore, FCS3.0 provides support for a cyclic redundancy check (CRC) word to be placed at the end of each FCS3.0 data set. The use of a CRC check word allows errors, occurring during file transfer or reading, to be detected. Time is increasingly used as a measurement parameter. Therefore, keyword support has been added to better describe the acquisition of time. Internationalization in the field of flow cytometry has caused a need for the incorporation of international characters in keyword values. Therefore, a provision has been made to support the use of multi-byte characters for some strings by providing a keyword to support the UNICODE character set (3).

Table of Contents

1. General
1.1 Scope
1.2 Conformance
2. Terminology and General Requirements
2.1 Conventions
2.2 Definitions
2.3 General Concepts
3. File Segments
3.1 HEADER Segment
3.2 TEXT Segment
3.3 DATA Segment
3.4 ANALYSIS segment
3.5 CRC Value
3.6 Other Segments
4. References
5. Appendices
5.1 Appendix A - Differences from FCS2.0
5.2 Appendix B - Data File Standard Committee Members
5.3 Appendix C - Proposed API for reading and writing FCS files

1. General

1.1 Scope

This is version 3.0 of the Flow Cytometry Data File Standard (FCS3.0). Its purpose is to provide detailed specifications for the structure of the data sets produced as a result of acquiring data on a cytometer and writing the data to a file.

1.2 Conformance

To be conformant with FCS3.0, a data file must conform to the file structure as described in this document and must contain all required keyword-value pairs in the primary TEXT segment of the file. A conformant file must not contain other segments not described in the data set HEADER segment. To be conformant with FCS3.0 an analysis program must be able to correctly read and interpret all of the data contained in any minimum FCS3.0 conformant file (a minimum FCS3.0 conformant file is one with only the required keyword-value pairs in the TEXT segment of the file and no information in the ANALYSIS segment).

2. Terminology and General Requirements

2.1 Conventions

2.1.1 The ASCII character code is used for all keywords and most of the keyword values throughout an FCS3.0 file (see section 3.2.20 regarding the use of UNICODE characters).

2.1.2 Numerical values are base 10 unless otherwise specified.

2.2 Definition

2.2.1 An FCS3.0 data file consists of one or more data sets.

2.2.2 A data set is defined as the collection of information produced by a cytometer as it carries out its measurements on some number of particles.

2.2.3 The collection of information in a data set is divided into at least four segments including a HEADER segment, a primary TEXT segment, a DATA segment, and an ANALYSIS segment. The ANALYSIS segment may be empty and any number of implementor-defined segments may follow the first four segments. New to FCS3.0 is the inclusion of an optional supplemental TEXT segment.

2.2.4 The HEADER segment identifies the data set as FCS3.0 and contains ASCII byte offsets from the start of the data set to the beginning and end of each of the other segments.

2.2.5 A keyword is the label of a data field. A keyword-value pair is the label of the data field with its associated value.

2.2.6 The TEXT segments contain a series of ASCII keyword-value pairs that describe the format of the DATA segment and most of the experimental operating conditions. The primary TEXT segment contains all required keyword-value pairs. The supplemental TEXT segment contains optional keyword-value pairs only.

2.2.7 The DATA segment contains either a list of the events or histograms of the data.

2.2.8 An event is an ordered list of the cytometric measurements for one particle. The length of an event is the number of parameters involved in the measurement.

2.2.9 A parameter is the signal produced by one of the detectors of the cytometer. Forward scattering is typically one of the measurement parameters. A parameter value is a digital representation of a parameter.

2.2.10 Each data set in a data file contains all the information needed to read and interpret the data set.

2.2.11 All space within a file which is not contained in a segment specified in the HEADER must be filled with a space character (ASCII 32). This includes unused space between the end of one segment and the beginning of the next segment and between the end of the last data set and the end of the file.

2.2.12 List mode data storage means that events are stored one after the other in a list.

2.2.13 All byte offsets are referenced to the beginning of the data set. The first data set in a file begins at byte zero of the file.

2.2.14 The implementor is the entity that creates the software to read and write FCS conformant data files.

2.2.15 The "delimiter" is the first character of the primary TEXT segment and is subsequently placed in the primary TEXT, supplemental TEXT and ANALYSIS segments to separate keywords from keyword values. The delimiter can be any ASCII character.

2.3 General Concepts

An FCS3.0 file is composed of one or more data sets, each containing at a minimum HEADER, TEXT and DATA segments. The HEADER, TEXT, and ANALYSIS segments contain ASCII-encoded text readable by a text editor (some keyword-values may contain UNICODE characters; See section 3.2.20). The DATA segment contains flow cytometry data stored in list mode or as histograms.

3. File Segments

3.1 HEADER Segment

3.1.1 The primary purpose of the HEADER segment is to describe the location of the other segments in the data set. The HEADER segment begins at byte offset zero from the beginning of the data set. The first six bytes in the HEADER segment comprise the version identifier (FCS3.0). Note, there is no space character between the FCS and the 3.0 in the identifier. The next 4 bytes (6 - 9) are occupied by space characters (ASCII 32). Following the identifier are at least three pairs of ASCII-encoded integers indicating the byte offsets for the start and end of the primary TEXT segment, the DATA segment, and the ANALYSIS segment, respectively. The byte offsets are referenced to the beginning of the data set. Under FCS3.0 these offsets remain limited to 8 bytes. Each ASCII encoded integer offset is right justified in its 8 byte space. The first byte offset (bytes 10 - 17) is that to the start of the primary TEXT segment. The next byte offset (bytes 18 - 25) is that for the end of the primary TEXT segment. The next offset (bytes 26 - 33) is that for the start of the DATA segment. The byte offset for the end of the DATA segment occupies bytes 34 - 41. That for the start of the ANALYSIS segment occupies bytes 42 - 49. The byte offset for the end of the ANALYSIS segment is in bytes 50 - 57. If there is no ANALYSIS segment these last two byte offsets can be set to zero (right justified) or left blank (filled with space characters). Offsets to the start and end of user-defined OTHER segments of the data set follow the ANALYSIS segment offsets. The user-defined segments will not be interpretable by others unless appropriate information is passed on by the data set originator.

A major change from previous FCS versions is the allowance for data sets larger than 99,999,999 bytes. When any portion of a segment falls outside the 99,999,999 byte limit, '0's are substituted in the HEADER for that segments begin and end byte offset. The byte offsets for begin DATA, end DATA, begin ANALYSIS, end ANALYSIS (begin and end supplemental TEXT if appropriate) will then only be found as keyword-value pairs in the primary TEXT segment. Note, when a segment is contained completely within the first 99,999,999 bytes of a data set, the byte offsets for that segment will be duplicated in the TEXT segment as keyword values. Note also, if the ANALYSIS offsets in the HEADER are zero, the $BEGINANALYSIS and $ENDANALYSIS keywords must be checked to determine if an ANALYSIS segment is present.

Table 1. Contents of HEADER fields and the byte offsets to the beginning and end of each field. Each offset is right justified in its field.

Contents	Start and end byte positions
FCS3.0	00 - 05
ASCII(32) - space characters	06 - 09

ASCII-encoded offset to first byte of TEXT segment 10 - 17

ASCII-encoded offset to last byte of TEXT segment 18 - 25

ASCII-encoded offset to first byte of DATA segment 26 - 33

ASCII-encoded offset to last byte of DATA segment 34 - 41

ASCII-encoded offset to first byte of ANALYSIS segment 42 - 49

ASCII-encoded offset to last byte of ANALYSIS segment 50 - 57

ASCII-encoded offset to user defined OTHER segments 58 - beginning of next segment

One example HEADER segment is as follows:

```
FCS3.0*********256****1545****1792**202456*******0*******0
```

The '*' character is used to represent a space character here. The TEXT segment starts at byte 256 from the location of the 'F' in FCS3.0 and ends at byte offset 1545. The DATA segment starts at byte offset 1792 and ends at 202456. There is no ANALYSIS segment, so the start and end offsets are shown as zeros. They could be left blank. Note that the HEADER segment is a continuous byte stream with no return or line feed characters. The bytes between the end of the HEADER segment and the start of the next segment must be filled with the space character. In this example, the segments are in the order HEADER, TEXT, DATA, and ANALYSIS. The FCS standard requires only that the HEADER segment be at the start of the data set and the primary TEXT segment be located entirely within the first 99,999,999 bytes.

A second example of a legal HEADER segment is:

```
FCS3.0*********256****1545*******0*******0*******0*******0
```

The '0's in the begin DATA and end DATA positions indicates that the DATA segment exceeds the 99,999,999 byte limit. Therefore, the byte offsets to begin Data and end Data, are located only in the $BEGINDATA, $ENDDATA keyword values in the TEXT segment. The begin ANALYSIS and end ANALYSIS byte offsets are also located in the $BEGINANALYSIS and $ENDANALYSIS keyword values in TEXT segment, if an ANALYSIS segment exist.

A third example of a legal HEADER segment is:

```
FCS3.0******202451**203140****1792**202450*******0*******0
```

This HEADER is different from the other examples in that it describes a data set in which the primary TEXT segment follows the DATA segment.

3.2 TEXT Segment

3.2.1 The TEXT segments (primary and supplemental) contain a series of ASCII encoded keyword-value pairs that describe various aspects of the data set. For example, $TOT/5000/ is a keyword-value pair indicating that the total number of events in the file is 5000. $TOT is the keyword and 5000 is the value. The '$' character flags this keyword as an standard FCS keyword. In this example, the '/' is the delimiter character.

3.2.2 A data set must contain a primary TEXT segment which contains all required keyword-value pairs and any number of optional keyword-value pairs. The primary TEXT segment must be contained entirely in the first 99,999,999 bytes of data set.

3.2.3 A data set may contain an optional supplemental TEXT segment that can contain only optional keyword-value pairs and may be placed anywhere in a data set after the HEADER segment.

3.2.4 The byte offset to the beginning and end of the supplemental TEXT segment is found in the $BEGINSTEXT and $ENDSTEXT keyword-value pairs which must be located in the primary TEXT segment.

3.2.5 The first character in the primary TEXT segment is the ASCII delimiter character. This character must also be used as the delimiter in the ANALYSIS and supplemental TEXT segments.

3.2.6 The delimiter is placed at the start and end of a keyword value.

3.2.7 The delimiter may not be the first character in a keyword or keyword value. If the delimiter appears in a keyword or keyword value, it must be immediately followed by a second delimiter. For example, "$SYS/RSX-11//M/" shows a value of RSX-11/M for the keyword $SYS. Since null (zero length) keywords or keyword values are not permitted, two consecutive delimiters can never occur between a value and a keyword.

3.2.8 All keywords are encoded in ASCII. Keyword values are encoded in ASCII by default. The values of specified keywords may be in languages not representable in ASCII by use of the $UNICODE keyword.

3.2.9 Keywords and keyword values must have lengths greater than zero.

3.2.10 Keywords are case insensitive, They may be written in a file in lower case, upper case, or a mixture of the two. However, an FCS file reader must ignore keyword case. A keyword value may be in lower case, upper case or a mixture of the two. Keyword values are case sensitive.

3.2.11 There are no default values for any keyword.

3.2.12 FCS-defined keywords must begin with the '$' character. Only FCS-defined keywords may begin with the '$' character.

3.2.13 FCS-defined keywords may not be redefined by the implementor.

3.2.14 There are required and optional FCS keyword-value pairs. The required keyword-value pairs represent the minimum set needed to successfully read and write an FCS data set. Conformant FCS file reading programs must recognize required FCS keywords.

3.2.15 The TEXT segments must not contain return (ASCII 13), line feed (ASCII 10) or other unprintable characters unless they are within a keyword value or are used as the delimiter character.

3.2.16 The parameter description keywords (e.g. $PnR, $PnB, etc) are numbered consecutively in the order in which the parameters are written to the file, beginning with number 1.

The required and optional FCS keywords are listed below with one line descriptions. The keywords and their values are described in alphabetical order following the lists. Required keywords are so indicated.

3.2.18 The required FCS primary TEXT segment keywords are as follows:

$BEGINANALYSIS Byte-offset to the beginning of the ANALYSIS segment.

$BEGINDATA Byte-offset to the beginning of the DATA segment.

$BEGINSTEXT Byte-offset to the beginning of a supplemental TEXT segment.

$BYTEORD Byte order for data acquisition computer.

$DATATYPE Type of data in DATA segment (ASCII, integer, floating point).

$ENDANALYSIS Byte-offset to the end of the ANALYSIS segment.

$ENDDATA Byte-offset to the end of the DATA segment.

$ENDSTEXT Byte-offset to the end of a supplemental TEXT segment.

$MODE Data mode (list mode, histogram).

$NEXTDATA Byte offset to next data set in the file.

$PAR Number of parameters in an event.

$PnB Number of bits reserved for parameter number n.

$PnE Amplification type for parameter n.

$PnR Range for parameter number n.

$TOT Total number of events in the data set.

3.2.19 The optional FCS TEXT segment keywords are as follows:

$ABRT Events lost due to data acquisition electronic coincidence.

$BTIM Clock time at beginning of data acquisition.

$CELLS Description of objects measured.

$COM Comment.

$COMP Fluorescence compensation matrix.

$CSMODE Cell subset mode, number of subsets to which an object may belong.

$CSVBITS Number of bits used to encode a cell subset identifier.

$CSVnFLAG The bit set as a flag for subset n.

$CYT Type of flow cytometer.

$CYTSN Flow cytometer serial number.

$DATE Date of data set acquisition.

$ETIM Clock time at end of data acquisition.

$EXP Name of investigator initiating the experiment.

$FIL Name of the data file containing the data set.

$GATE Number of gating parameters.

$GATING Specifies region combinations used for gating.

$GnE Amplification type for gating parameter number n.

$GnF Optical filter used for gating parameter number n.

$GnN Name of gating parameter number n.

$GnP Percent of emitted light collected by gating parameter n.

$GnR Range of gating parameter n.

$GnS Name used for gating parameter n.

$GnT Detector type for gating parameter n.

$GnV Detector voltage for gating parameter n.

$INST Institution at which data acquired.

$LOST Number of events lost due to computer busy.

$OP Name of flow cytometry operator.

$Pkn Peak channel number of univariate histogram for parameter n.

$PKNn Count in peak channel of univariate histogram for parameter n.

$PnF Name of optical filter for parameter n.

$PnG Amplifier gain used for acquisition of parameter n.

$PnL Excitation wavelength for parameter n.

$PnN Short name for parameter n.

$PnO Excitation power for parameter n.

$PnP Percent of emitted light collected by parameter n.

$PnS Name used for parameter n.

$PnT Detector type for parameter n.

$PnV Detector voltage for parameter n.

$PROJ Name of the experiment project.

$RnI Gating region for parameter number n.

$RnW Window settings for gating region n.

$SMNO Specimen (tube or well) label.

$SRC Source of the specimen (patient name, cell types)

$SYS Type of computer and its operating system.

$TIMESTEP Time step for time parameter.

$TR Trigger parameter and its threshold.

$UNICODE UNICODE code page for string type keyword values.

3.2.20 Alphabetical listing and detailed description of keywords. For all the keywords below 'n', 'n1', 'n2', etc represent ASCII-encoded integer values. The character 'f' represents an ASCII-encoded floating point number. The word "string" represents an ASCII or UNICODE-encoded TEXT string that can be of any length greater than zero. If the optional $UNICODE keyword is used, a specified subset of the strings may be represented with two byte characters in a variety of UNICODE conformant languages. Otherwise strings are in single byte ASCII. The character 'c' represents a single ASCII-encoded character. The '/' character is used here as the delimiter for illustrative purposes.

$ABRT/n/ $ABRT/1265/

Number of events lost due to data acquisition electronic coincidence effects. The number of aborted events here was 1265.

$BEGINANALYSIS/n/ $BEGINANALYSIS/123456789/ [REQUIRED]

This field contains the byte-offset from the beginning of the data set to the beginning of the optional ANALYSIS segment. If there is no ANALYSIS segment, a '0' should be placed in this keyword value. In this example, the ANALYSIS segment begins at byte 123,456,789.

$BEGINDATA/n/ $BEGINDATA/123456789/ [REQUIRED]

This field contains the byte-offset from the beginning of the data set to the beginning of the DATA segment. If the DATA segment is completely contained within the first 99,999,999 bytes of the data set, this value duplicates the offset contained in the HEADER segment. In this example, the DATA segment begins at byte 123,456,789

$BEGINSTEXT/n/ $BEGINSTEXT/123456789/ [REQUIRED]

This field contains the byte-offset from the beginning of the data set to the beginning of the supplemental TEXT segment. If there is no supplemental TEXT segment, the value should be set to '0'. In this example, the supplemental TEXT segment begins at byte 123,456,789.

$BTIM/hh:mm:ss[:tt]/ $BTIM/14:22:10:47/

Clock time at the beginning of data acquisition. The format of the value is 24-hour clock hours:minutes:seconds:number of fractional seconds in units of 1/60 of a second. The fractional seconds [:tt] is optional. Data acquisition began at 14 hours, 22 minutes, 10 seconds, and 47/60th of a second.

$BYTEORD/n1,n2,n3,n4/ $BYTEORD/4,3,2,1/ [REQUIRED]

This keyword specifies the order from numerically least significant[1] to numerically most significant[4] in which four binary data bytes are written to compose a 32-bit word in the data acquisition computer. The numbers are separated by commas (ASCII 44). In VAX computers and personal computers of the IBM PC type, the byte order is 1,2,3,4 with the least significant byte written first. In Hewlett Packard, Macintosh and Sun computers, the byte order is 4,3,2,1 meaning that the least significant byte is written last. In PDP-11 computers the byte order is 3,4,1,2 meaning that in the two 16-bit words comprising a 32-bit word, the most significant 16-bit word is written first. Within the 16-bit word, however, the least significant byte is written first, which is the same as for a PC. Byte order is discussed more fully in reference 4. In this example, the most significant byte is written first and the least significant byte is written last. Use of this keyword enables collection of data on one computer type and analysis of the data on another computer type.

$CELLS/string/ $CELLS/Normal human peripheral blood/

Type of cells or other objects measured. This specimen is normal human peripheral blood.

$COM/string/ $COM/Incubation time was 47 minutes./

This keyword is used to attach a comment to the data set. It should not to be used as a substitute for other standard keywords. This example shows the use of **$COM** to add a brief note to the data set, a note that otherwise might appear only in a laboratory notebook.

$COMP/n,f1,f2,f3,.../ $COMP/3,0.0,-0.1,0.0,-40.0,0.0,-0.6,0.0,-36.4,0.0/

This keyword enables the efficient storage of a fluorescence compensation matrix. The matrix has n rows and n columns where n represents the number of acquisition parameters. f1, f2,

f3, ... are floating point numbers representing the matrix elements. Both positive and negative values are allowed. A positive or unsigned value indicates that compensation has been additive while a negative value indicates the more common case of subtractive compensation. The elements are stored in row-major order, i.e., the elements in the first row appear first. The matrix element Cij is the percentage of FLj that has been subtracted electronically from FLi. In the example, the compensation matrix is 3 x 3 and the matrix elements have the following subtractive values: C11=0.0%, C12 = 0.1%, C13 = 0.0%, C21 = 40.0%, C22=0.0%, C23 = 0.6%, C31 = 0.0%, and C32 = 36.4%, C33 = 0.0%.

$CSMODE/n/ $CSMODE/3/

Cell subset mode, i.e., the number "n" of subsets to which a object may belong. The simplest case is that the cell subset parameter encodes a single value per object as would be indicated by n = 1. If the value of n is greater than 1 it indicates that the value of the cell subset parameter may encode n subset identifiers. In these cases, the

$CSVBITS and **$CSVnFLAG** keyword values will specify how the cell subset values are encoded. It should be noted that regardless of the value for this keyword, a cell subset value of zero indicates that the object is undefined by the analysis scheme that was used.

$CSVBITS/n/ $CSVBITS/4/

The number of bits used to encode a cell subset value. When the **$CSMODE** keyword value is greater than 1, the number of bits used to encode a cell subset identifier must be specified by the **$CSVBITS** keyword value. In the cited example, 4 bits, i.e., values of 0-15, are used to encode cell subset identifiers. See the discussion of the ANALYSIS segment in section 3.4.

$**CSVnFLAG $CSV1FLAG/4096/**

The value used as a "flag" to indicate that the "n" identifier field encodes a value. In the cited example, if bit 13 is set in the value of the cell subset parameter (parameter value AND 8192 is TRUE), one should read the second field of bits to decode the value. It is not necessary to set "flags", but if one wishes to use zero to encode the first subset for any field, one must set a "flag" to indicate that the zero in that field refers to a subset. See the discussion of the ANALYSIS segment in section 3.4.

$CYT/string/ $CYT/FACScan/

The name of the flow cytometer used for the data set. Here a FACScan was used.

$CYTSN/string/ $CYTSN/400E370/

The serial number of the flow cytometer used for the data set. Here the serial number is 400E370.

$DATATYPE/c/ $DATATYPE/I/ [REQUIRED]

This keyword describes the type of data written in the DATA segment of the data set. The four allowed values are 'I', 'F', 'D', or 'A'. The DATA segment is a continuous bit stream with no delimiters. 'I' stands for unsigned binary integer, F stands for single precision IEEE floating point, 'D' stands for double precision IEEE floating point, and 'A' stands for ASCII. The additional keywords **$PnB** (bits per parameter) and **$PnR** (range per parameter) are needed to completely describe an event in the DATA segment.

$DATATYPE/I/** means that the events are written as unsigned binary integers. For each parameter in an event, both the maximum length in bits allocated for storage of the parameter and the actual integer range used by the parameter within that allocation are needed. The number of bits per parameter is specified by **$PnB. For example, **$P1B/16/** specifies that 16 bits are allocated for parameter 1. **$P1R/1024/** specifies that parameter 1 values range from 0 to 1023. This allows the data word length to be specified, facilitating compatibility between machines with different data word lengths and enabling bit compression of the data.

$DATATYPE/F/** means that the data are written as single precision floating point values in the IEEE standard format. Note that the **$PnB keywords should be set to a value of 32 for each parameter in an event. For example, **$P1B/32/**.

$DATATYPE/D/** means that the data are written as double precision floating point values in the IEEE standard format. The **$PnB keyword should be set to a value of 64 for each parameter in an event. For example, **$P3B/64/** says that parameter 3 is allocated 64 bits of storage space. The IEEE standard formats for single- and double-precision numbers are given in the table below:

<u>Single-precision Double-precision</u>

Sign bit 31 bit 63

Exponent bits 30-23 bits 62-52

bias 127 bias 1023

Fraction bits 22-0 bits 51-0

Range 3.402823e+38 1.797693e+308

approx. 1.175494e-38 2.225074e-308

$**DATATYPE/A/** means that the data are written as ASCII-encoded integer values. In this case, the keyword **$PnB** specifies the number of bytes allocated per value (one byte per character). This represents fixed format ASCII data. **$P1B/4/** indicates that the maximum value for parameter 1 would be 9999. Data are stored in a continuous byte stream, with no delimiters. If the value of the **$PnB** keyword is the * character, e.g., **$P1B/*/**, the data are free format and number of characters per parameter value may vary. In this case, all values are separated by one of the following delimiters: "space", "tab", "comma", "carriage return", or "line feed" characters. Note that multiple, consecutive delimiters are treated as a single delimiter. Since there are significant differences between the way in which consecutive delimiters are treated by different programming languages, care should be taken when using this format. Zero values must be explicitly specified by the zero (0) character. Thus, the string "1,3,, ,3" (note the space between the third and fourth commas) would only specify three values. It would be treated as between 3 and 5 values by different programming languages.

$DATE/dd-mmm-yyyy/ $DATE/01-OCT-1994/

This keyword specifies the date on which the data set was created. The format is day-month-year with the number of characters specified by dd-mmm-yyyy. This data set was created on 01 October 1994. Note that the all the character positions should be filled including leading zeros. Accepted abbreviations for the months are: JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC.

$ENDANALYSIS/n/ $ENDANALYSIS/123456789/ [REQUIRED]

This field contains the byte-offset from the beginning of the data set to the end of the ANALYSIS segment. If there is no ANALYSIS segment, a '0' should be placed in this keyword value. In this example, the ANALYSIS segment ends at byte 123,456,789

$ENDDATA/n/ $ENDDATA/123456789/ [REQUIRED]

This field contains the byte-offset from the beginning of the data set to the end of the DATA segment. If the DATA segment is completely contained in the first 99,999,999 bytes of the data set, this value duplicates the offset contained in the HEADER segment. In this example, the DATA segment ends at byte 123,456,789.

$ENDSTEXT/n/ $ENDSTEXT/123456789/ [REQUIRED]

This field contains the byte-offset from the beginning of the data set to the end of the supplemental TEXT segment. If there is no supplemental TEXT segment, the value should be set to '0'. In this example, the supplemental TEXT segment ends at byte 123,456,789.

$ETIM/hh:mm:ss[:tt]/ $ETIM/14:22:10:47/

Clock time at the end of data acquisition. The format of the value is 24-hour clock hours:minutes:seconds:number of fractional seconds in units of 1/60 of a second. Data acquisition ended at 14 hours, 22 minutes, 10 seconds, and 47/60th of a second. The fractional seconds keyword value is optional as indicated by the square brackets.

$EXP/string/ $EXP/A. Smith/

The name of the person initiating the experiment. This experiment was under the direction of A. Smith.

$FIL/string/ $FIL/071494.001/

The name of the data file that corresponds to this data set. If there is only one data set in the FCS file, then this file name should be the same as the name of the FCS file. If this data set is one of several in the FCS file, then the file name may correspond to a data file collected at some earlier time. In this example, the data are stored in a file named 071494.001.

$GATE/n/ $GATE/2/

This keyword specifies the number of parameters used for gating. It is analogous to the **$PAR** keyword, which specifies the total number of parameters for each event in the data set. In this example, there are two gating parameters. The current practice in many flow cytometry laboratories is that the gating parameters are collected as part of the data set. This fact is reflected in the redefinition of the **$RnI** keyword described below.

$GATING/string/ $GATING/R1/ $GATING/R1 AND (R2.0R.R3)/

This keyword specifies the conditions under which the data in the data set have been acquired. The conditions are set through Boolean operations among regions defined below using the **$RnI** and $**RnW** keywords. Allowed Boolean operators are AND, OR(inclusive), and NOT. The operands are the regions (Rn). Operators are separated from operands or other operators by spaces or periods. Operator precedence is from left to right unless overridden with parentheses. In the first example, data were collected using gating region R1. Events with parameter values falling outside R1 were excluded from the data set. In the second example, an event is included in the data set only if the appropriate parameter value is inside R1 and is inside R2 or R3 or both.

$GnE/f1,f2/ $G3E/4.0,0.01/

This keyword specifies whether linear or logarithmic amplifiers were used for gating parameter number n. When the amplification is logarithmic the value of f1 specifies the number of logarithmic decades and f2 represents the linear value that would have been obtained for a signal with a log value of 0. In the example above, the data for parameter 3 were collected using a four-decade logarithmic amplifier and the 0 channel represents the linear value, 0.01. When linear amplification is used or when amplification is undefined such as with some calculated parameters, f1 and f2 are set to 0.

$GnF/string/ $G2F/520LP/

This keyword specifies the optical filter that was used for the light reaching the detector for gating parameter n. This example shows that the optical filter used for the second gating parameter was a type 520 nm long pass.

$GnN/string/ $G1N/FL2/

This keyword specifies a short name for gating parameter number n. Here "FL2" is the name for gating parameter 1. Required short names for parameters include the following:

FS Forward Scatter

SS Side scatter

FLn Fluorescence channel n

AE Axial Extinction

CV Coulter Volume

TIME Time

$GnP/n1/ $P3P/27/

The amount of light collected by the detector for gating parameter number n1 expressed as a percentage of the light emitted by a fluorescent object. In the example, 27% of the emitted light was captured by the detector for gating parameter number 3.

$GnR/n1/ $G2R/1024/

This keyword specifies the range, n1, of gating parameter n. In this example, the events for gating parameter 2 range from 0 to 1023.

$GnS/string/ $G1S/FITC-CD45/

This keyword specifies a longer name for gating parameter n than is allowed by **$GnN**. Here, FITC-labeled CD45 is the name for gating parameter 1.

$GnT/string/ $G2T/PMT9524/

This keyword specifies the detector type for gating parameter n. Here, gating parameter 2 uses a photomultiplier tube (PMT) of type 9524.

$GnV/n1/ $G2V/645/

This keyword specifies the detector bias voltage for gating parameter n. In this example, the detector for gating parameter 2 is biased at 645 volts.

$INST/string/ $INST/Laboratory of FCM, RPCI/

The institution or laboratory in which the data were collected. In this example, the data were collected in the Laboratory of Flow Cytometry at Roswell Park Cancer Institute.

$LOST/n/ $LOST/457/

This keyword specifies the number of events lost during data acquisition because the computer was busy with other tasks. Here, 457 events were so lost.

$MODE/c/ $MODE/L/ [REQUIRED]

This keyword specifies the mode in which the data were acquired. Allowed values for the character c are 'C', 'L', or 'U'. These options are described as follows:

C One correlated multivariate histogram is stored in the data set as a multidimensional array. There can be only one such histogram per data set. In storing multiparameter correlated data, the index for the first parameter is incremented first, then the second, etc. For bivariate data, the first data value corresponds to index 1 for parameter 1 and index 1 for parameter 2, the second data value corresponds to index 2 for parameter 1 and index 1 for parameter 2, etc.

L List mode. For each event, the value of each parameter is stored in the order in which the parameters are described. The number of bits reserved for parameter 1 is described using the **$P1B** keyword. There can be only one set of list mode data per data set. The **$DATATYPE** keyword describes the data format. This is the most versatile mode for the storage of flow cytometry data because mode C and mode U data can be created from mode L data.

U Uncorrelated univariate histograms. There can be more than one univariate histogram per data set. The histogram frequencies for parameter 1 are stored first followed by those for parameter 2, etc. If the univariate histograms have been gated, they must all have been acquired with the same gates so that the total number of events in each histogram is the same.

$NEXTDATA/n/ $NEXTDATA/202512/ [REQUIRED]

When there is more than one data set in an FCS file, this keyword gives the byte offset from the beginning of a data set to the first byte in the HEADER of the next data set in the FCS file. If n is zero (0), this is the final or only data set in the file. This example shows that the next data set begins at byte 202512 from the beginning of the present data set. Each data set stands alone and must contain a full complement of keywords.

$OP/string/ $OP/Dave/

The name of the operator of the flow cytometer. Here Dave was the operator of this instrument.

$PAR/n/ $PAR/5/ [REQUIRED]

This keyword specifies the total number of parameters stored in each event in the data set. In this example, data for five parameters are stored for each event.

$Pkn/n1/ $PK2/374/

For a univariate histogram of parameter n, this keyword specifies the channel number, n1, containing the highest frequency of events. In this example, the peak in the univariate histogram for parameter 2 is located in channel 374. The **$PKNn** keyword specifies the count in that channel.

$PKNn/n1/ $PKN2/12803/

For a univariate histogram of parameter n, this keyword specifies the number of events, n1, in the channel number (histogram bin) containing the maximum event frequency. In this example, the univariate histogram for parameter 2 has a maximum event frequency of 12803. The **$PKn** keyword above specifies that this peak count occurs at channel 374.

$PnB/n1/ $P3B/16/ [REQUIRED]

For $**DATATYPE/I/**(binary integers), this keyword specifies the number of bits allocated, n1, for storage of parameter n. In this example, the data value for parameter 3 would be stored as two bytes (16 bits). This keyword is used in conjunction with **$PnR** to determine how the data are actually stored. A flow cytometer with 10-bit analog-to-digital converters (ADCs) would have **$PnR/1024/.** A 10-bit number would be stored in the 16-bit space allocated by **$PnB/16/** leaving 6 empty bits per parameter. These keywords enable tight bit packing of events. For example, the data storage could be specified by **$PnB/10/$PnR/1024/** for each of the n parameters in an event. Then fewer bits would be wasted in storing each event. However, packing these data for storage and unpacking them later for analysis is very time-consuming. In practice, most flow cytometers use **$PnB/16/$PnR/1024/** for 10-bit data. A flow cytometer with 8-bit ADCs would use **$PnB/8/$PnR/256/** where n represents integers from one to the number of parameters measured.

For **$DATATYPE/A/**(ASCII-encoded integers), **$PnB** specifies the number of characters, n, per measured value for parameter n.

$PnE/f1,f2/ $P3E/4.0,0.01/ [REQUIRED]

This keyword specifies whether linear or logarithmic amplifiers were used for parameter number n. When the amplification is logarithmic the value of f1 specifies the number of logarithmic decades and f2 represents the linear value that would have been obtained for a signal with a log value of 0. In the example above, the data for parameter 3 were collected using a four-decade logarithmic amplifier and the 0 channel represents the linear value, 0.01. When linear amplification is used or when amplification is undefined such as with some calculated parameters, f1 and f2 are set to 0.

$PnF/string/ $P2F/520LP/

This keyword specifies the optical filter that was used for the light reaching the detector for parameter n. This example shows that the optical filter used for the second parameter was a type 520 nm long pass.

$PnG/f/ $P2G/10.0/

This keyword specifies the gain that was used to amplify the signal for parameter n. This example shows that parameter 2 was amplified 10.0-fold before digitization.

$PnL/n1/ $P1L/488/

This keyword specifies the excitation wavelength, n1, in nm for parameter n. In this example, the wavelength was 488 nm for parameter number 1.

$PnN/string/ $P3N/FL1/

This keyword is used to specify the short name of parameter n. Here parameter 3 has a short name of FL1. Required short names for parameters include the following:

CS Cell subset

FS Forward Scatter

SS Side scatter

FLn Fluorescence channel n

AE Axial Extinction

CV Coulter Volume

TIME Time

$PnO/n1/ $P2O/200/

This keyword specifies the excitation power, n1, in milliwatts for the light source associated with the measurements for parameter n. Here 200 mW was used to produce the signal associated with parameter 2.

$PnP/n1/ $P4P/50/

The amount of light collected by the detector for parameter number n expressed as a percentage of the light emitted by a fluorescent object. In the example, 50% of the emitted light was captured by the detector for parameter number 4.

$PnR/n1/ $P2R/1024/ [REQUIRED]

This keyword specifies the maximum range, n1, of parameter n. For **$MODE/L/** (list mode data), this corresponds to the ADC range, here 1024. The data values can range from 0 to 1023. For univariate histogram data (**$MODE/C/** or **$MODE/U/**), it is the number of channels, n1, in the histogram for parameter n. Here the histogram channel numbers range from 0 to 1023.

$PnS/string/ $PnS/CD45 FITC Fluorescence/

This keyword specifies a long name to be used as an axis label in a plot of parameter n. Here FITC-labeled CD45 is the label. **$PnS** is the long name equivalent of **$PnN**.

$PnT/string/ $P2T/PMT9524/

This keyword specifies the detector type for parameter n. Here, parameter 2 uses a photomultiplier tube (PMT) of type 9524.

$PnV/n1/ $P2V/645/

This keyword specifies the detector bias voltage, n1, in Volts for parameter n. In this example, the detector for parameter 2 is biased at 645 Volts.

$PROJ/string/ $PROJ/AML patient study/

This keyword provides the name of the project. Here it is an AML patient study.

$RnI/string1,[string2]/ $R3I/P2,P4/ $R2I/G3/

This keyword associates a gating region number, n, with one or two parameters, here shown as string1 and string2. The two strings are of the form "Pn" or "Gn". "Pn" stands for collected parameter n, while "Gn" stands for gating parameter n. In the first example, gating region 3 is associated with a bivariate dot plot or bivariate histogram for parameters 2 and 4. The **$RnW** keyword described below specifies the shape of the gating region. In the second example, gating region 2 is associated with gating parameter 3. See the discussion for the **$GATE** keyword.

$RnW/n1, n2[;n3, n4;...]/ $R1W/345, 366/

This keyword specifies the window settings for gating region n. This window setting is useful only if the **$RnI** keyword is also specified. If the keyword **$RnI** has only a single value, then n1 and n2 specify the inclusive lower and upper bounds for the window in a univariate histogram. For example, **$R2I/3/$R2W/345,366/** specifies that gating region 2 is associated with gating parameter 3. The gated events must range between channels 345 and 366 inclusive. If the **$RnI** keyword value has two values, then the window exists in a bivariate plot and it is specified in the **$RnW** keyword as a polygon. The x and y coordinates of the first point in the polygon are the pair n1, n2. The next point is separated from the first by a ';' character and is represented as n3, n4 above. The polygon can contain any number of points separated by semicolons. The first point and the last point are assumed to be connected. For example, **$R1I/2,3/$R1W/310,205;515,304;480,615;240,514;354,542/** specifies that region 1 is defined in parameter 2 and 3 and that the region 1 window is a 5-sided polygon in this 2-parameter space. The **$GATING** keyword will specify the way the windows will be used (AND, OR, etc.).

$SMNO/string/ $SMN0/A7/

This keyword specifies the specimen number, which could be a tube or well number. Here the specimen number is A7.

$SRC/string/ $SRC/J. Doe, HIV positive patient/

This keyword specifies the source of the specimen. Note that this keyword value could contain patient information, which is protected by the U.S. Privacy Act and by strict U.S. National Institutes of Health guidelines. The acquiring laboratory may choose to use encoded information for this keyword value.

$SYS/string/ $SYS/Macintosh System 7.5/

This keyword specifies the type of computer and the operating system under which the data set was collected. Here the data set was collected on a Macintosh running System 7.5.

$TIMESTEP/f/ $TIMESTEP/0.0167/ $TIMESTEP/1.0/

The presence of this keyword indicates that time has been collected as one of the parameters in the data set. **$PnN/TIME/** specifies which parameter represents time. **$TIMESTEP** specifies the time step in seconds. In the first example, the time step is 0.0167 seconds, which is 1/60 of a second and is the typical clock tick on a personal computer. For this example, an implementor specifies **$P6N/TIME/$P6B/16/ $P6R/65536/$TIMESTEP/0.0167/**. When the first event in the data set is captured by the computer, the number of clock ticks since the computer was turned on is read and saved as a constant, n Ticks. A zero value is entered into parameter 6 in this first event. When the second event arrives, the number of clock ticks is obtained from the computer clock. n Ticks is subtracted from this number and the result stored as parameter 6 of the second event. The actual number of seconds between any subsequent event and the first event is obtained by multiplying the parameter 6 value by the **$TIMESTEP** value. In this example, the maximum time range is approximately 17.5 minutes. In the second example, an implementor specifies **$P6N/TIME/$P6B/16/$P6R/65536/$TIMESTEP/1.0/**. Using the same procedure as in the first example, any events arriving less than 1.0 second after the first event have a parameter value of zero, while those arriving between 1.0 second and (less than) 2.0 seconds have a parameter 6 value of 1. The maximum time range is approximately 18 hours. If an external constant time interval generator is used to provide a signal input that increases linearly with time, the appropriate TEXT keywords might be **$P6N/TIME/$P6B/16/$P6R/1024/**

$TIMESTEP/0.001/. Here the time step is smaller than that available from the computer clock. However, the number of steps is limited by the range of the ADC, here 10 bits. The maximum time range for this example is 1023 seconds.

$TOT/n/ $TOT/25000/ [REQUIRED]

This keyword specifies the total number of events in the data set. This data set contains 25000 events.

$TR/string, n/ $TR/FS,54/

This keyword specifies the parameter name which serves as the trigger signal for an event. The number, n, is the channel number of the threshold signifying an event. When the threshold is exceeded, an event is declared. Here forward scatter (FS) is the trigger signal and the event threshold is at channel 54.

$UNICODE/n,string1,string2,etc/ $UNICODE/3,$SYS,$SRC/

The integer 'n' represents the UNICODE page number used and the comma delimited strings represent the keyword values where UNICODE text is used. UNICODE is an international standard that enables computer representation of most of the world's languages. The characters for each language are represented as two-byte codes on a code page. There are 65536 codes available. U.S. ASCII requires 256 two-byte characters. For computer systems that support UNICODE, implementors will be able to present axis labels and other appropriate text strings in the language of the country in which the flow cytometry data are being collected. If this keyword is not present, single byte U.S. ASCII is used for all strings. In the example above, UNICODE page 3 was used to write the values for the **$SYS** and **$SRC** keywords.

3.3 DATA Segment

The DATA segment contains the raw data in one of three modes (list, correlated or uncorrelated) described in the primary TEXT segment by the **$MODE** keyword value. The data are written to the DATA segment in one of four allowed formats (binary, floating point, double precision floating point or ASCII) described by the **$DATATYPE** keyword value. The most common form of data storage is list mode storage in the form of binary integers **($DATATYPE/I/ $MODE/L/**). The **$PnB** set of keywords specify the bit width for the storage of each parameter. The **$PnR** set of keywords specify the channel number range for each parameter. For example, **$P1B/16/ $P1R/1024/** specifies a 16-bit field for parameter 1 and a range for the values of parameter 1 from 0 to 1023, which corresponds to 10 bits. Implementors should use a bit mask when reading these list mode parameter values to insure that erroneous values are not read from the 4 unused bits.

3.4 ANALYSIS segment

ANALYSIS is an optional segment that, when present, contains the results of data processing. It is often the case that analysis is performed off-line, after the data has been collected and stored in a data set. Therefore, the ANALYSIS segment typically contains information added to a copy of the original file. For examples, the results of cell cycle analysis or immunophenotype determinations often involve more complex analyses than can be performed in "real time" as the data is collected and stored. The ANALYSIS segment has the same structure as the TEXT segment; i.e., it consists of a series of keyword-value pairs. There are no required keywords for the ANALYSIS segment. The optional FCS keywords are listed in 3.4.1 with one line descriptions and in 3.4.2 with full descriptions and examples. Implementors may add their own keywords.

A proposal has been made that the ANALYSIS segment be used for identifying cell subsets, determined either by region drawing or by some partitioning method such as cluster analysis (4). This may be particularly useful for immunophenotyping data. Three approaches to identifying cell subsets are discussed below. The first two use the least space in the data set but require the cell subsets be disjoint. The third approach adds a parameter to each event and supports overlapping cell subset assignments.

In method 1, the implementor uses the TEXT segment keyword-value pairs **$CSMODE/1**/ and **$CSTOT/n/** to specify that there is one group of cell subsets containing n disjoint subsets of cells. The TEXT segment keyword-value pair **$CSVBITS/8/** is used to indicate that the cell subset assignments for each event are stored in a binary vector of unsigned characters (8 bits each) whose length is the number of events in the data set. This vector is stored in an other segment following the ANALYSIS segment. The DATA segment contains a copy of the original data with the events written in the same order as in the original data set. In the ANALYSIS segment, **$CSnNUM** is used to count the number of cells in each of the n subsets.

In method 2, the implementor uses the TEXT segment keyword-value pairs **$CSMODE/1/** and **$CSTOT/n/** as above but does not use the **$CSVBITS** keyword. In the DATA segment, the events are written out one cell subset at time rather than in the original event order. In the ANALYSIS segment, **$CSnNUM** is used to count the number of cells in each of the n subsets. No other segment is required.

Method 3 creates an additional cell subset (CS) parameter for each event in the data set. Cell subsets may be defined by the method, e.g., cluster analysis, neural network, boolean gates on combinations of parameters, hyperplanes in n-dimensional space, etc. The value of the parameter may encode a single subset identifier number for each event (**$CSMODE/1/**) or more than one identifier number per event (value of **$CSMODE** greater than 1). The meanings of the identifier numbers are specified by the values of the **$CSnNAME** keywords in the TEXT and ANALYSIS segments. If the value of the CS parameter is 0 (zero), that event is unclassified by the definitions used to assign cell subsets. If the classification scheme creates unique non-overlapping populations, e.g., CD4 T cells, CD8 T cells, B cells, monocytes/macrophages, neutrophils, etc., then the simplest approach is to set the value of **$CSMODE** to "1" and use 1==CD4 T cell, 2 == CD8 T cells, etc. In some situations, it may be useful to be able to assign a single cell to more than one defined subset. For example, to extend the preceding example, subset identifiers 1 - 5 would correspond the definitions listed above with 6 == lymphocytes and 7 == mononuclear cells. This scheme would require **$CSMODE/3/** since a single cell could belong to three defined subsets. Operationally, assuming that an event in the data set is a CD4 T cell, then the first bit field would encode a value of 1 (CD4 T cell), the second bit field would encode a value of 6 (lymphocyte), and the third bit field would encode a value of 7 (mononuclear cell). The bit fields and their interpretations in these cases would be defined by the values of the **$CSVBITS** and the **$CSVnFLAG** keywords as outlined in the reference (4). Method 3 also supports the creation of an ANALYSIS segment that includes a summary for the results written as the values for the keywords pertaining to the numbers of cells in each subset, etc. Method 3 has the size "cost" of an additional parameter, but it permits one to include a complete and explicit record of an analysis as an integral part of a data set.

3.4.1 Optional FCS ANALYSIS segment keyword list:

$CSDATE Cell subset analysis date.

$CSDEFFILE Cell subset definition file name.

$CSEXP Name of person who performed the cell subset analysis.

$CSnName Name of cell subset number n.

$CSnNUM Number of cells in cell subset number n.

3.4.2 Optional FCS ANALYSIS segment keywords:

$CSDATE/dd-mmm-yyyy/ $CSDATE/26-OCT-94/

Cell subset date. This keyword specifies the date on which the data set containing the cell subset analysis was created. The format is of the date is the same as that for **$DATE**. This data set was created on 26 October 1994.

$CSDEFFILE/string/ $CSDEFFILE/c:\filename.dat/

Cell subset definition file. The string is the name of the file containing the information needed to define each of the cell subsets. In the example the cell subset definition file is named filename.dat and is located on drive c:.

$CSEXP/string/ $CSEXP/A. Smith/

Cell subset experimenter. Name of the person who performed the cell subset analysis. Here, A. Smith performed the cell subset analysis.

$CSnName/string/ $CS2N/lymphocytes/

Cell subset name. This is a string naming cell subset number n. In the example, cell subset 2 is named "lymphocytes".

$CSnNUM/n1/ $CS2NUM/3456/

This keyword specifies the number of cells, n1, in cell subset number n. In the example, cell subset 2 contains 3456 cells.

3.5 CRC Value

The CRC word is computed for the part of each data set beginning with the first byte of the HEADER segment and ending with the last byte of the final segment of the data set (which could be a TEXT, DATA, ANALYSIS or OTHER segment). The CRC word is a 16-bit cyclic redundancy check value (5). This 16-bit CRC word conforms to the CCITT standard (Comite' Consultatif International Te'le'graphique et Te'le'phonique).

This standard uses the CCITT polynomial X16 + X12 + X5 and requires that each input character be interpreted as its bit-reversed image. These requirements are satisfied by the icrc function in reference 6 if the last two function arguments are 0 and -1, respectively. The CRC value will be placed as ASCII in the 8 bytes immediately after the end data set. If an implementor chooses not to compute and store a CRC word then the 8 bytes immediately after the end of the data set should be filled with ASCII '0' characters.

3.6 Other Segments

Implementors may create any number of OTHER segments as they choose.

4. References

1. Murphy RF, Chused TM:A proposal for a flow cytometric data file standard. Cytometry 5:553-555, 1984.

2. Dean PN, Bagwell CB, Lindmo T, Murphy RF and Salzman GC: Data File Standard for Flow Cytometry. Cytometry 11:323-332, 1990.

3. The Unicode Consortium: The UNICODE Standard, Version 1.0, vol. 1. Addison-Wesley Publishing Co. Inc., Reading, MA, 1991.

4. Redelman D, Coder DM: Cell subset (CS) parameter to record the identities of individual cells in flow cytometric data. Cytometry 18:95-102, 1994.

5. Press WH, Teukolsky SA, Vetterling WT, Flannery BP: Numerical Recipes in C. 2nd ed. Cambridge University Press, Cambridge, UK, 1992.

5.1 Appendix A: Major Differences Between FCS2.0 and FCS3.0.

1) The HEADER has been modified to accommodate data sets longer than 99,999,999 bytes. Any offset value that requires more than 8-bytes is now represented by placing a '0' in the HEADER for that value and its associated "$BEGIN" value. The actual byte-offset value is then found in the primary TEXT segment of the data set. This system allows the vast majority of data files to be backwards compatible with analysis software designed for previous FCS versions. However, a '0' byte-offset in the HEADER will prevent previous FCS versions from reading very large data sets, avoiding read errors or partial data reads. Note, $BEGINDATA, $ENDDATA, $BEGINANALYSIS, $ENDANALYSIS, $BIGINSTEXT and $ENDSTEXT keyword-value pairs are required in the HEADER segment of FCS3.0 conformant files irrespective of the size of the data set. When the size of a data set remains below the 100 megabyte limit, the byte offsets will be found both in the HEADER and in keyword value pairs in the primary TEXT segment. When a data set reaches or exceeds 100 megabytes, byte offsets will only be located in the primary TEXT segment.

2) A supplemental TEXT segment may now be included in a data set. The supplemental TEXT segment may contain only optional keyword-value pairs and may be located anywhere in a data set after the HEADER segment.

3) A primary TEXT segment must contain all required keyword-value pairs and be located entirely within the first 99,999,999 bytes of a data set.

4) An optional 16-bit CRC check has been added to the end of each data set. This internal check-word allows for data set integrity checks.

5) To enable third party or off-line analysis software to correctly read and interpret data, the keyword $PnE is now required for each parameter. The $PnE keyword describes the method of amplification used for a given parameter.

6) There are a number of new optional FCS TEXT Segment keywords. $CSMODE, $CSTOT, $CSVBITS, $CSVnFLAG specify an added parameter to identify cell subsets. $CYTSN specifies the cytometer serial number. $RnI has been redefined. $TIMESTEP has been added to enable use of a time parameter. $UNICODE enables the specification of certain keywords in languages not representable with ASCII text.

7) The $DATE keyword value for year is increased by two bytes to -yyyy.

8) The following optional ANALYSIS segment keywords have been added: $CSDATE, CSDEFFILE, $CSEXP, $CSnN, and $CSnNUM to enable specification of cell subsets.

9) The definition of the $BYTEORD and $PnE keywords have been corrected and clarified. The $PnG keyword has been added, describing the linear gain applied to a signal.

10) The $COMP keyword has replaced $DFCiTOj for the description of fluorescence compensation.

5.2 Appendix B: Data File Standards Committee of the International Society for Analytical Cytology

Larry Seamer, Chair
Director, Flow Cytometry Facility
University of New Mexico
Cancer Center, Cytometry
900 Camino de Salud NE
Albuquerque, NM 87131
(505) 277-6206
lseamer@cobra.unm.edu

Bruce Bagwell
Maine Medical Center Research Institute
70 John Roberts Road, Suite 8
South Portland, ME 04106
75450.167@compuserve.com

Luther Barden
Div. of Computer Research and Technology,
Building 12A Room 2015
National Institutes of Health
9000 Rockville Pike
Bethesda, MD 20892
luther_barden@nih.gov

Marc Christofferson
Becton Dickinson Immunocytometry Systems
2350 Qume Drive
San Jose, California 95131-1807
(408) 954-2058
[N.B.: Marc Christofferson is no longer with BDIS.]

Louise E. Magruder
Division of Clinical Laboratory Devices
FDA/CDRH/ODE
72 Gaither Road
Rockville, MD 20850
lem@fdadr.cdrh.fda.gov

George Malachowski
Cytomation, Inc.
400 E. Horsetooth Rd.
Ft. Collins, CO
(303)226-2200

Robert F. Murphy
Associate Professor
Department of Biological Sciences and Center for
Light Microscope Imaging and Biotechnology
Carnegie Mellon University
4400 Fifth Avenue, Box 52
Pittsburgh, Pennsylvania 15213
(412) 268-3480
murphy+@cmu.edu

Doug Redelman
Sierra Cytometry
3150 Susileen Dr.
Reno, NV 89509

Gary C. Salzman
Life Sciences Division
Los Alamos National Laboratory
Mail Stop M888
Los Alamos, NM 87545
(505)667-5503
salzman@lanl.gov

James C.S. Wood
Coulter Corporation
Mail Code 52-A01
11800 S.W. 147th Avenue
Miami, FL 33196-2500
(305)380-2449 or 344-1290 (voice)
(305)344-5240 (FAX)
woodjcs@gate.net

� 1996 International Society for Analytical Cytology