Cytometry 11:323–332 (1990)

Data File Standard for Flow Cytometry

Data File Standards Committee of the Society for Analytical Cytology

Received for publication November 27, 1989; accepted November 27, 1989

This data file standard for flow cytometry (FCS) provides a detailed description of a data file structure designed such that the file can include all of the information necessary to describe fully: 1) the instrument used to obtain the data; 2) the sample measured; 3) the data obtained; and 4) the results of analysis of the data. The file may contain one or more data sets. Each data set includes data from a single sample or acquisition run. Each data set consists of a minimum of four parts—HEADER, TEXT, DATA, and ANALYSIS—all of which are required. The file is structured such that the parts of a data set do not have to be in any particular order, except for the HEADER, which must always be first. The HEADER contains pointers to the beginning and end of each of the other three parts. The file is written as a continuous byte stream, with no line or page formatting. The HEADER, TEXT, and ANALYSIS parts are written in ASCII format and the DATA part can be written in ASCII, binary integer or floating point.

This Standard, noted as FCS2.0, is based on a standard proposed by Murphy and Chused (*Cytometry* 5:553–555, 1984). Data written utilizing their proposed format (FCS1.0) is compatible with the format described here. Future versions of the standard will maintain compatibility with older versions.

GENERAL PRINCIPLES

1. A data set contains all of the information relative to a flow cytometric measurement.
2. Data from more than one measurement may be stored in a single file, as a number of data sets.
3. Each data set contains four mandatory parts (HEADER, TEXT, DATA, and ANALYSIS) and any number of additional (user-defined) parts.
4. Each data set is complete unto itself, not dependent on any other data set in the data file.
5. The HEADER is always the first part of a data set. It contains the byte address of the beginning and end of all other parts of a data set, relative to the beginning of the data set. This feature means that the parts do not have to be contiguous or in a particular order.
6. The TEXT part of a data set contains *KEYWORDS* and their associated *VALUES*. The form of a keyword value depends on the nature of the keyword (e.g., character string or number). There is **NO** default for any keyword. Keywords and their values may be of any length (number of bytes) except that they must contain at least one character (no nulls). Keywords are case (upper/lower) insensitive; values are case sensitive.
7. FCS keywords begin with the "$" character. They are recognized by all programs with formats as specified in the Standard. User-defined keywords must begin with some other character.
8. There are *required* and *optional* keywords. The required keywords contain all of the information necessary to read the data part of the data set. Optional keywords are used to supply additional information about the measurement.
9. All space within a file which is not being used will be filled with a space character.

SPECIFIC FILE PARTS

HEADER

The HEADER part of each data set in a data file begins with the Flow Cytometry Standard version identifier, which occupies the first 10 bytes. This version of the Standard will be entered as "FCS2.0" followed by four spaces. The next eight (8) bytes contain the offset, in bytes, from the beginning of the data set to the start of the TEXT part of the data set; i.e., an offset of 60 bytes points to the 61st byte of the data set. All offsets are in ASCII, right justified. The next eight (8) bytes contain the offset, again in bytes, from the beginning of the data set to the last byte of the TEXT part. The next four (4) 8-byte numbers give the offsets to the beginning and end of the DATA part, and the beginning and end of the ANALYSIS part. If any part of the data set is not included in the file, the offsets can be zero (0) or blank. Additional pairs of locally defined offsets may follow the standard three (3) pairs if desired. Thus, the first 58 bytes of the data set are of fixed format and contain information that identifies the version number of the Standard used and pointers to the beginning and end of all mandatory parts of the data set. The offsets contained in the HEADER part of a second data set refer to the first byte of the second data set; i.e., all offsets within a data set are relative to the first byte of the data set. In addition to the standard offsets described above, which point to the standard parts of a data set, a user can add additional offsets to point to user-defined parts of the data set. These user-defined parts will not be readable by another laboratory, unless the user passes the appropriate information along to the laboratory.

TEXT

The TEXT part of a data set consists primarily of KEYWORD and keyword VALUE combinations. The *first* byte of the TEXT part defines a "separator" character. The separator character is inserted after each keyword and after each keyword value. For example, for "$TOT/10000/" the keyword is $TOT, the separator is the "/" character, and the value of $TOT is 10000. The separator character may not be the first character in a keyword or in a keyword value. If the separator appears in a keyword or in a keyword value, it must be "quoted" by being repeated. For example, "$SYS/RSX-11//M/" shows a value of RSX-11/M for the keyword $SYS. Since null (zero length) keywords or keyword values are not permitted, two consecutive separators can never occur between a value and a keyword. The remainder of the TEXT part consists of repeats of "KEYWORD separator VALUE separator", etc. Although standard keywords begin with "$", it is not necessary to "quote" the "$" to include it within either a KEYWORD or KEYWORD VALUE, since null keywords are not allowed (and therefore "$" only has significance in the first position of a keyword name). The standard keywords and the format of their values are described below. Note that some of the keywords are mandatory. In addition, the user will find it advantageous to include sufficient keywords to describe clearly the data set. The TEXT part should not contain "carriage return" or "line feed" characters unless they are within a keyword value or unless used as the separator.

This standard describes a predetermined set of keywords. These keywords are reserved for the uses described and are not to be redefined by the user. Users may define other keywords for their own use.

DATA

The keyword $DATATYPE defines the format of the data. It has four possible values: **I, F, D, and A.**

$DATATYPE = I. The data are written as unsigned binary integers. The length in bits of the data values is specified by the keyword "**$P*n*B**". Here, the letter "**P**" stands for "Parameter", or the variable being measured, the letter "**B**" stands for *bits,* and the letter "***n***" stands for the number of the parameter being specified (e.g., ***n*** = 1 might be for forward light scatter and ***n*** = 2 for fluorescence). This allows the data word length to be specified dynamically, facilitating compatibility between machines with different data word lengths and/or allowing bit compression of the data. Data are stored in a continuous stream of bits, with no delimiters.

$DATATYPE = F. 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.

$DATATYPE = D. The data are written as double precision floating point values in the IEEE standard format. Note that the $PnB keywords should be set to a value of 64.

	Single-Precision	Double-Precision
Sign	bit 31	bit 63
Exponent	bits 30–23 bias 127	bits 62–52 bias 1023
Fraction	bits 22–0	bits 51–0
Range, approx.	3.402823e + 38 1.175494e-38	1.797693e + 308 2.225074e-308

$DATATYPE = A. The data are written as ASCII-encoded integer values. In this case, the keyword "**$P*n*B**" specifies the number of bytes per value (one byte per character). This represents fixed format ASCII data. For example, for **$P1B = 6**, the maximum value would be 999999. Data are stored in a continuous byte stream, with no delimiters. If the value of this keyword is the * character (e.g., **$P1B**/*/), the number of characters per value may vary. In this case, all values will be separated by delimiters: "space", "tab", "comma", "carriage return", and "line feed" characters are allowed delimiters. 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 signified 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.)

For all types of data storage, and for all modes of data storage (i.e., for all values of the keyword **$MODE**), the order in which the parameters are written to the file must be the same as the order of specification. For list mode data (**$MODE = L**, to be described later), the first parameter value stored in the data part of the data set must be described by the **$P1*** set of keywords, the second parameter value by the **$P2*** keywords, etc. For uncorrelated single-parameter histograms (**$MODE = U**), the histogram values for parameter 1 must be stored first, then histogram values for parameter 2, etc.

In storing multiparameter correlated data, the index for the first parameter is incremented first, then the

FCS 2.0 STRUCTURE

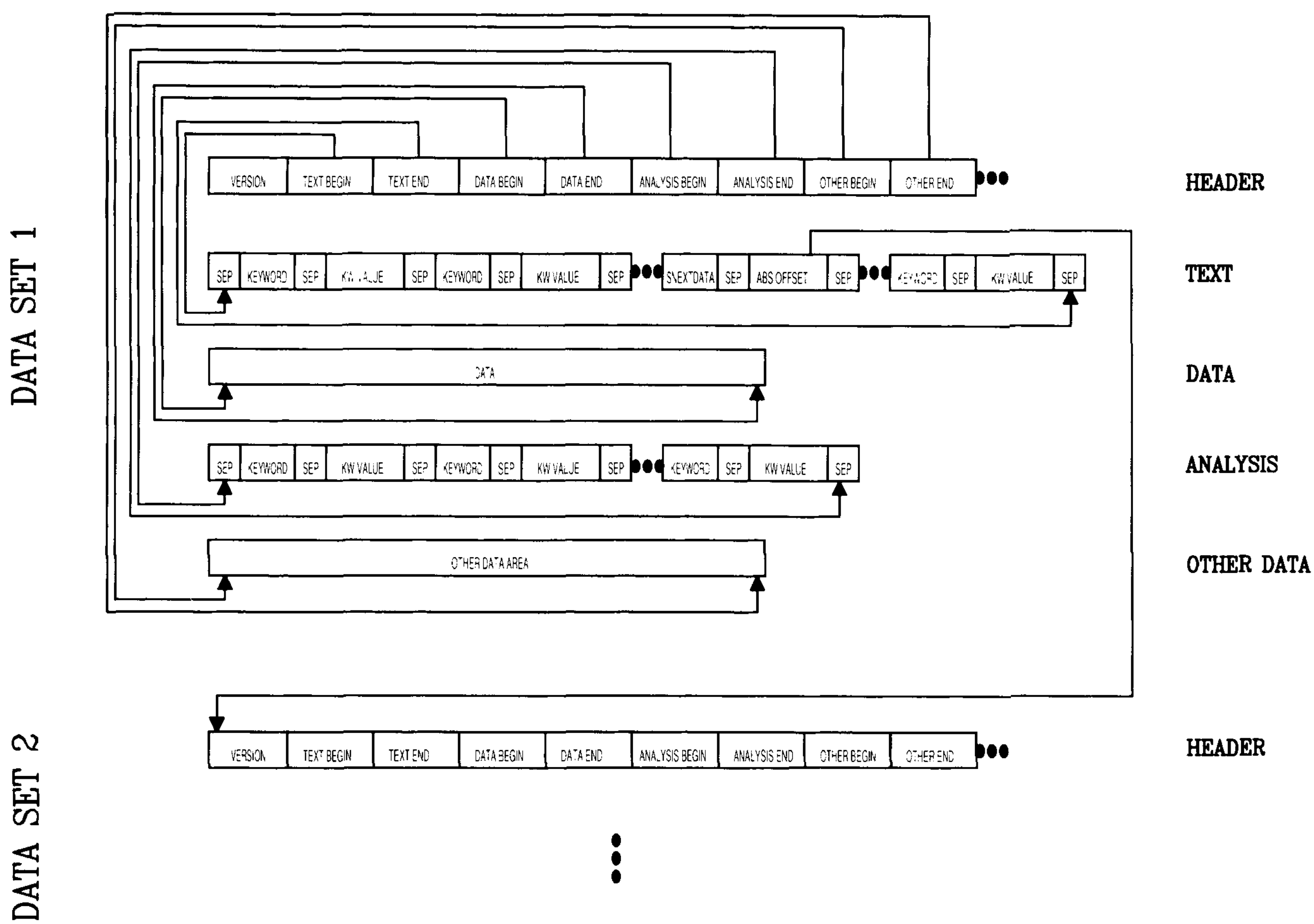

FIG. 1. The structure of an FCS 2.0 Standard Data File. All parts of the file are included, with arrows indicating the way the byte offsets point to the beginning and end of each part. In the HEADER part, the OTHER BEGIN and OTHER END offsets show how the user can add information to the file, information not required by the Standard. In the TEXT part, the offset labelled ABS OFFSET points to the beginning of a second data set that is stored in the file.

second, etc. For two-parameter data, this means that 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. For data sets containing values longer than eight bits (more than one byte), the keyword $BYTEORD specifies the order, from low to high, in which the bytes have been stored. For example, the value "1,2,3,4" indicates that the lowest-order byte is stored first (e.g., for the Digital Equipment Corporation VAX family of computers). The value "2,1,4,3" indicates that the *high*-order *byte* of each 16-bit word is stored first, but that the *low*-order 16-bit *word* of a 32-bit value is stored first (e.g., for the Motorola 68000 series of computers).

A data set may contain any number of uncorrelated ($MODE = U) single-parameter histograms. If gated, they must have all been acquired with the same gates and thus contain the same number of events; see the $TOT keyword. However, it may contain only one correlated ($MODE = C, e.g., two-parameter). Also, a data set may contain only one data list ($MODE = L).

ANALYSIS

The ANALYSIS part of the data set would typically contain information added to the file after it was collected and stored. Examples are the results of cell cycle analysis or percent positive enumeration. The ANALYSIS part of a data set, if present, will have the same structure as the TEXT part; i.e., it will consist of a series of keyword/value combinations. FCS 2.0 does not define the keywords; they are left to the user.

Figure 1 illustrates the structure of an FCS 2.0 data file. Although the file would be written as a continuous bit stream, in this illustration it is separated into its parts. All parts of the file are included, with arrows indicating the way the byte offsets point to the beginning and end of each part. In the HEADER part, the OTHER BEGIN and OTHER END offsets show how the user can add information to the file, information not required by the Standard. In the TEXT part, the offset labelled ABS OFFSET points to the beginning of a second data set that is stored in the file.

STANDARD KEYWORDS AND THEIR DEFINITIONS

The following section describes the set of keywords currently supported by the Flow Cytometry Data File Standard. An example is given for each keyword to illustrate its use. In the examples, the separator between keywords and their values is the "/" character. In each case, the keyword is given a possible value and the combination is explained. In some cases, the keyword contains the "*" character, e.g., $P1*. The * character represents a number of options for the keywords; e.g., $PnB, $PnR, $P1N.

Many keywords include the character ***n***. Each parameter stored or used in creating a gated data set is assigned a number. This number is then used in all of the appropriate keywords used to describe the parameter. The $P1N, $P1B, $P1F, and $P1R keywords would describe features of the first parameter stored in a data set. This number may not be the same one assigned to a parameter by the user when the data are collected. For example, if three parameters are measured (DNA, RNA, forward scatter) one might call them parameters 1, 2, and 3, respectively. However, if one were to store in a data set only the second parameter (RNA), the number assigned and used in the $Pn* keywords would be 1 ($P1N, $P1B, etc.). If one were to store more than one single-parameter histogram in the data set, the second set of keywords ($P2B, etc.) would describe the features of the data stored for the second parameter. A third histogram would use $P3B, etc. There is no limit to the number of single-parameter histograms that can be stored in a data set.

A data set may contain only one gated histogram. In this case, the $Pn* keywords represent the gated data. An additional set of keywords, $Gm*, may be used (optional but recommended) to describe the parameters used in the gating. The "m" is the sequential number of the gate parameter being described. It is not the same as the "n" used with stored parameters; i.e., $P1B and $G1B do not describe the same parameter. Storage of the data used to create the gating conditions is also optional. If stored, however, they must be placed in a separate data set. Gating conditions are specified through another set of keywords—$GATE, $RjI, $RjW, and $GATING—where j is the gating region number.

The keywords are listed in two sections; required and optional. The basic principal is that *required* keywords are those necessary for reading the data part of the data set.

Required KEYWORDS

$DATATYPE	**$PAR**	**$MODE**	**$PnB**
$BYTEORD	**$PnR**	**$NEXTDATA**	

Optional KEYWORDS

These keywords are grouped according to function.

Specifications for the instrument and the sample

$DATE	**$EXP**	**$$PROJ**	**$OP**
$INST	**$FIL**	**$CYT**	**$SMNO**
$SRC	**$SYS**	**$CELLS**	
$BTIM	**$ETIM**		

Number of events

$TOT	**$LOST**	**$ABRT**

Data storage and gating specifications

$TR	**$DFCiTOj**	**$COM**	**$PKn**
$PKNn	**$RjI**	**$RjW**	**$GATING**
$Pn*	**$Gm***	**$GATE**	

Required KEYWORDS

$DATATYPE **e.g., $DATATYPE/A/**

This keyword specifies the format of the data stored in the DATA part of the data set. It replaces the $ASC keyword of FCS1.0. Four formats are defined, as discussed earlier.

- **I** Data are stored as unsigned binary integers.
- **F** Data are stored as 32-bit floating point numbers (IEEE standard).
- **D** Data are stored as 64-bit floating numbers (double precision).
- **A** Data are encoded in ASCII.

In the example, the data are stored in the ASCII format.

$MODE **e.g., $MODE/U/**

The **$MODE** keyword specifies the mode of the data. It can have several values:

- **U** Uncorrelated single-parameter histograms. There can be more than one single-parameter histogram in one data set.
- **C** One correlated multiparameter histogram. There can be only one of these histograms per data set. Two parameter data are stored in row/column format, with x stored first.

L List mode. For each event, the value of each parameter is stored in the order in which the parameters are described (i.e., with the $Pn* keywords).

The example shows that the data in this data set are uncorrelated. There may be more than one histogram in the data set.

$BYTEORD e.g., **$BYTEORD/1,2,3,4/**

This keyword specifies the order in which the data bytes are written, least to most significant, with the byte numbers separated by commas. As described in the description of the DATA part of the data set, the byte order varies with type of computer and its operating system. In this example, four bytes are stored in order, with the least significant byte first.

$NEXTDATA e.g., **$NEXTDATA/12300/**

This keyword gives the byte offset to the header of an additional data set included in a single file. If zero (0), there are no more data. This example shows that the next data set begins at byte 12300, relative to the first byte of the current data set.

$PAR e.g., **$PAR/8/**

This keyword specifies the total number of parameters (signals) stored in the data set. In this example, data for eight parameters are stored.

$P*n*B e.g., **$P1B/16/**

For data stored as binary images, this keyword specifies the number of bits stored for each value of parameter number ***n***. A value of 16 means that two bytes will be written for each data point, for parameter ***n***. For data stored as ASCII-encoded integers, it specifies the number of bytes (characters) per measured value. As discussed earlier, the parameter number defines the order in which parameters are written to the data set. The first parameter in this data set contains 16 bits per data value.

$P*n*R e.g., **$P2R/256/**

This keyword specifies the range of parameter n. For histogram data, it is the number of channels in the histogram. For list mode data, it is the total number of events stored. In this case, the number of data values is the number of cells times $PAR (the number of parameters per cell). In the example, the second parameter written to the data set has 256 channels in the histogram.

Optional KEYWORDS

These keywords are grouped by function.

$DATE e.g., **$DATE/10-JAN-89/**

The date the file was created. The file was created on 10-JAN-89; format is dd-mmm-yr. **Note:** This format should always be used for the date, to permit calculations of elapsed time in a standard fashion.

$EXP e.g., **$EXP/A. Smith/**

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

$PROJ e.g., **$PROJ/Ara-c drug resistance of CHO cells/**

This keyword is used to provide the name of the project. The name of the project under which the data were generated was "Ara-C drug resistance of CHO cells.

$OP e.g., **$OP/A. Smith/**

The name of the operator of the instrument. A. Smith was the operator of this instrument.

$INST e.g., **$INST/NCI, NIH/**

The institution (laboratory) where the data were collected. The data were collected within a laboratory of the NIH.

$FIL e.g., **$FIL/F2213.DAT/**

The name of the data file. This is provided as a cross check for proper storage of the data. The data are stored in a file named F2213.DAT.

$CYT e.g., **$CYT/FCM-1/**

The name of the cytometer used for the measurement. The cytometer used was called FCM-1.

$SMNO e.g., **$SMNO/23/**

The number of the sample measured. This entry can be used to correlate the data stored with an entry in a notebook. It could also be a tube or well number for automated sample handlers, e.g., A7. The sample number for this data set is 23.

$SRC e.g., **$SRC/B. Jones/**

The source of sample (patient's name, cell line, animal strain, etc.). This sample came from B. Jones.

$SYS e.g., **$SYS/RSX-11//M 4.1/**

The operating system of the computer to which the data were written. The operating system was RSX-11/M 4.1. Note that the "/" character had to be repeated to show that it was a part of the value of the keyword.

$CELLS **e.g., $CELLS/ Peripheral blood/**

Type of cells measured. This could also refer to other objects, e.g., chromosomes. This sample came from peripheral blood.

$BTIM **e.g., $BTIM/14:22:10/**

The time at the beginning of data collection; format hh:mm:ss. **Note:** This format should always be used for the time so that elapsed time calculations can be performed in a standard fashion. The collection of data began at 14:22:10 (2:22 P.M.).

$ETIM **e.g., $ETIM/14:27:10/**

Time at end of data collection (same format restrictions as **$BTIM**). Data collection ceased at 14:27:10 (2:27 P.M.).

$TOT **e.g., $TOT/100000/**

The number of objects for which data are stored in a distribution or list. Data from 100,000 objects is stored in this data set.

$LOST **e.g., $LOST/2154/**

The number of objects lost from the distribution due to the computer CPU being busy. In this sample, data were not stored for 2,154 objects due the CPU (computer) being busy when they passed through the instrument.

$ABRT **e.g., $ABRT/1539/**

The number of aborts, objects that were not counted due to coincidence effects in the electronics. The number of aborts for this sample was 1,539.

$TR **e.g., $TR/FS,8/**

When one signal is ued as the master trigger signal that defines an event (e.g., cell detected), this keyword is used to specify the name of the trigger signal, and its threshold setting. In this example, a forward scatter signal with a minimum size of 8 is used to trigger the remainder of the electronics, defining the presence of a measured cell.

$DFC*i*TO*j* **e.g., $DFC1TO3/5/**

Some dye combinations have emission spectra that overlap. This effect means that a part (usually small) of the signal from one dye may be added to the signal from the other dye. In some instruments it is possible to provide electronic compensation to correct for this effect. This example shows that electronic compensation is being used, from parameter **1** to parameter **3**. The correction is given in percent. In this case, 5% of the value of parameter **1** is subtracted from the value of parameter **3**.

$COM **e.g., $COM/This cell line donated by the CDC./**

This keyword is used to insert a comment into the data set. It is not to be used to circumvent the use of other standard keywords. This example shows the use of the keyword to add a brief note to the data set, a note that otherwise might appear only in a lab notebook.

$PK*n* **e.g., $PK2/25/**

Use this keyword to specify the peak channel number (i.e., the channel with the most counts in it) for variable ***n***. The peak in the distribution for parameter 2 is in channel 25.

$PKN*n* **e.g., $PKN2/12803/**

This keyword specifies the number of objects (e.g., cells) that are in the peak channel for parameter ***n***. There are 12,803 objects in the peak channel of the distribution for parameter 2.

$P*n**

This is a set of keywords where the * can have several forms. These keywords describe the parameters for the data stored in the data set. Two versions of this keyword are required; they are $PnB and $PnR which were described under the section on required keywords. Again, the ***n*** in these keywords corresponds to the order in which the parameters were stored in the data set.

$P*n*N **e.g., $P1N/FS/**

This keyword is used to specify the name of the parameter measured by each of the ***n*** detectors. In this example, parameter number 1 is forward light scatter. Possible values for the name are:

FS	Forward Scatter
SS	Side scatter
FL	Fluorescence
AE	Axial Extinction
CV	Coulter Volume
TI	Time

$P*n*S **e.g., $P2S/Anti-IgM/**

The name of the fluorescent stain or probe used with parameter ***n***. In the example, parameter 2 identifies cells marked with fluorescently labelled anti-IgM.

$P*n*F e.g., **$P2F/520LP/**

The F stands for filter. This example shows that the optical filter used for the second parameter was a type 520 nm long pass.

$P*n*E/j,k/** e.g., **$P1E/12,0.01/

This keyword specifies whether linear or logarithmic amplifiers were used. For logarithmic amplification, the value specifies the number of decades (**j**) and the offset (**k**). The offset is the value on a linear scale corresponding to channel 1 in the log histogram, as a fraction of full scale (0.0 to 1.0). For linear amplification, the number of decades is set to zero. This is an optional keyword with no defaults. This means that if the keyword is not present in the data set, no assumption can be made as to the type of amplifier used. In the example, a four-decade logarithmic amplifier was used with an offset of 0.01 with parameter 1.

$P*n*L e.g., **$P1L/488/**

This keyword specifies the wavelength of the laser used to acquire the data for parameter ***n***, in nanometers. In this example, the wavelength was 488 nm for parameter number 1.

$P*n*O e.g., **$P2O/200/**

Excitation output power in milliwatts, 200 mw in this example with parameter number 2.

$P*n*P e.g., **$P1P/50/**

The percent of emitted light collected for parameter ***n***. In the example, 50% of the emitted light was measured for parameter number 1.

$P*n*T e.g., **$P3T/PMT9524A/**

Photomultiplier tube or detector type. In the example, a model 9524A photomultiplier tube was used for parameter number 3.

$P*n*V e.g., **$P3V/800/**

Detector voltage, in this case 800 volts for the photomultiplier used to measure parameter number 3.

$GATE e.g., **$GATE/2/**

This keyword has the same meaning for gate parameters as the keyword **$PAR** has for stored parameters. It specifies the number of gating parameters described by the **$Gm*** keywords. The example shows that two gating parameters were used in the measurement being recorded.

$G*m**

These keywords are analogs of the **$Pn*** keywords, including **GmR**. They are used to describe the parameters used in gating. The ***m*** is the sequential number of the gate parameter being described. It is not necessarily the same as the ***n*** used with the **$P*n**** keywords. Since these keywords are not required, information about the gating parameters may not be stored at all. The ***m*** will be used again when defining the gating conditions. These keywords have exactly the same options listed above for the * character and for the values.

The next three keywords (**$R*j*I, $R*j*W and $GATING**) are used to specify the gating parameters and the gating conditions.

$R*j*I

This keyword defines gating regions, where the ***j*** is the gating region number. Its value gives the number(s) of the gating parameter(s) used to establish the gating region. If the keyword has a single value, then the region is defined in a single-parameter histogram of that parameter number; e.g., **$R1I/2/** says that gating region 1 is defined in the data from gating parameter 2, which is described by the **$G2*** keywords. If the keyword has two values separated by a comma, then the gating region is defined in a two-parameter histogram; e.g., **$R2I/2,3/** shows that the gating region 2 is defined in a histogram of gating parameters 2 and 3.

$R*j*W

This keyword specifies the window settings for gating region **j**. The value contains the coordinates of the window. If **$RjI** contains a single value, then the window will be specified in a single-parameter histogram. **$R1I/1/$R1W/50,75/** specifies a single window, channels 50 to 75 inclusive, in gating parameter 1.

$R1I/2,3/$R1W/10,20;10,40;30,40;30,20/ specifies a polygonal window in gating parameters 2 and 3. In this example the window is a square but it can have any shape, in which case all of the vertices of the polygon would be included, separated by semicolons. The vertices must be listed in sequence around the region. The region will be generated by connecting the vertices in the order in which they are stored. The polygon will be implicitly closed from the first to the last vertex if the first vertex is not repeated at the end of the list. More examples of the use of this keyword are presented in the EXAMPLES section of this document. The **$GATING** keyword will specify the way the windows will be used (AND, OR, etc.).

$GATING **e.g., $GATING/R1/**

This keyword specifies the gating conditions. It defines which gating regions are used and, through the use of operators, how they interact. The example shows the simplest type of gating; the data stored in the data set were obtained by using gating region number 1 (R1). More than one region could be used: **$G1N/FS/$R1I/1/$R2I/1/$R1W/10,20/$R2W/30,40/ $GATING/R1 OR R2/** specifies two regions, both in gating parameter 1, defined as forward light scatter by the keyword **$G1N/FS/**. The **OR** operator says that an event will be stored only if it has a value between 10 and 20 or between 30 and 40 (inclusive) in the gating parameter (FS). Operators that are possible with this keyword are:

AND OR NOT

Operators must be separated from operands or other operators by spaces or by periods. Operators are executed from left to right, unless modified through the use of parentheses: e.g.,

$GATING/R1 AND (R2.OR.R3)/

Parentheses may also be used to provide clarity in an expression: e.g.,

$GATING/R2.AND.(.NOT.R3)/

Users are free to add their own keyword and keyword value combinations, using a leading character other than the $. However, the keywords will be private to the user and general purpose software written to read and process the data file may not recognize them.

EXAMPLES

Several examples are given using most if not all of the options. The separator character in these examples is the "/" character.

F	C	S	2	.	0											6	0						1	2	9				
	1	3	0						6	4	1																		
/	$	M	O	D	E	/	U	/	$	P	A	R	/	1	/	$	D	A	T	A	T	Y	P	E	/	I	/	$	P
1	B	/	1	6	/	$	B	Y	T	E	O	R	D	/	1	,	2	/	$	P	1	R	/	2	5	6	/	$	N
E	X	T	D	A	T	A	/	0	/																				

Although the information shown above is printed on five lines, in the file it would be in one continuous byte stream. The information is displayed in this form so that the spacing is clear. The measurement data would follow the last "/" character. The first 58 bytes is the **HEADER** part of the file. The first 10 bytes, "**FCS2.0** ", show that Version Number 2.0 of the Standard was used in creating the file. They are followed by a series of eight-byte words that give the offsets from the beginning of the data set to each of its parts. These offsets always refer to the first byte of the data set. The first word (eight bytes) shows that the **TEXT** part of the file begins 60 bytes past the beginning of the data set, " **60**"; i.e., on the 61st byte. The **TEXT** part ends 129 bytes past the beginning of the data set, " **129**". The next 16 bytes show that the **DATA** part of the file starts at an offset of 130, " **130**", and ends at an offset of 641, " **641**", for a total of 512 bytes (two bytes per data point). The next 16 bytes show that this file has no **ANALYSIS** part; all 16 bytes are blank. Since offsets are used to show where the different parts of a data set are located in the file, the header part can be extended to include user-defined offsets to other parts of the file. Note that the **HEADER** part contains only 58 bytes while the **TEXT** part begins with an offset of 60; there are two blank bytes between them. The first byte of the **TEXT** part of the file is the "/" character, the separator. The first keyword/value combination (**$MODE/U/**) shows that uncorrelated single-parameter data are stored in the data set. The next keyword (**$PAR/1/**) shows that data are stored for only one parameter. The third keyword (**$DATATYPE/I/**) shows that the data are written as binary integers. The fourth keyword (**$P1B/16/**) shows that 16 bits (two bytes) are stored for each data point of parameter 1. The fifth keyword (**$BYTEORD/1,2/**) show that the least significant byte is written first. The next keyword (**$P1R/256/**) specifies that the histogram contains 256 channels (512 bytes). The final keyword (**$NEXTDATA/0/**) shows that there are no more data sets in the file. The last separator character is followed by the 512 bytes of histogram data. This data set contains only the *required* keywords. Others could be added, either standard ones or keywords defined by the user. Although the extra keywords would logically follow the **$P1R** keyword, they could in fact be placed anywhere within the **TEXT** part of the file. If additional keywords were to be added, the byte offsets in the **HEADER** part of the file would have to be changed.

In the following examples, only the TEXT part of the data set is shown, with only the required keywords.

1. Store two uncorrelated (single-parameter) histograms.

$MODE/U/$PAR/2/$DATATYPE/I/$P1B/16/$P1R/ 64/$P2B/16/$P2R/ 256/$BYTEORDER/1,2/ $NEXTDATA/0/

This TEXT definition specifies uncorrelated data (**$MODE/U/**) with 64 16-bit words (**$P1B/16/$P1R/64/**)

containing a histogram for parameter 1 and 256 16-bit words (**$P2B/16/$P2R/256**) containing a histogram for parameter 2. There are two parameters (**$PAR/2/**) and the data are stored as binary integers (**$DATATYPE/I/**).

2. Store a list of data of four parameters.

$MODE/L/$PAR/4/$DATATYPE/I/$P1B/16/$P1R/1024/ $P2B/16/$P2R/1024/$P3B/16/$P3R/256/ $P4B/16/$P4R/256/$BYTEORD/1,2/ $TOT/10000/ $NEXTDATA/0/

This TEXT definition specifies a total of 10,000 (**$TOT/10000/**) measurements stored in a list (**$MODE/L/**). All four parameters are stored as 16-bit words (**$PnB/16/** for n = 1 to 4). The range of values for parameters 1 and 2 is 1,024 channels (**$P1R/1024/$P2R/1024/**), which means that only the lowest 10 bits of these data words contain the data. The range for parameters 3 and 4 is 256 channels (**$P3R/256/$P4R/256/**), which means that only the lowest eight bits (least significant byte) of these data words contain data. The total number of bytes written to the data part of this data set is (10,000 cells) × (four parameters per cell) × (two bytes per parameter) = 80,000.

Note that combinations such as **$P1B/7/$P2B/9/** are possible, in which case significant computation might be required to read and write the data, depending on the instruction set of the computer being used. The purpose for using such an arrangement would be to save disk space. In this example, two bytes store two parameters (total of 16 bits). In the more conventional format, three bytes would be required; one for parameter 1 and two for parameter 2. Thus the disk space required for the data specification shown would be reduced by a third.

GATING

The following examples illustrate how gating can be specified by using the Standard. Note: The HEADER is not included in these examples, nor are all of the required keywords.

Assume four parameters are measured:

1. Forward Scatter (FS)
2. Side Scatter (SS)
3. DNA (FL1)
4. RNA (FL2)

1. A histogram is to be stored for DNA gated on windows in both FS (20–30) and RNA (30–60). There are two independent windows, not one window in FS vs RNA.

   $P1N/FL1/$P1S/DNA/$GATE/2/$G1N/FS/ $G2N/FL2/$G2S/RNA/$R1I/1/$R1W/20,30/ $R2I/2/$R2W/30,60/$GATING/R1 and R2/

   The **$GATING** keyword uses the **AND** operator to show that for an event to be written to the data set there must be signals in the windows of gating parameters 1 *and* 2. Note that gating parameter 2 is RNA, not SS. The ordering of gating parameters (**$Gm***) is independent of the ordering of the stored parameters (**$Pn***) and of the acquired parameters.

2. Store DNA signals when an event is detected in either of two gating windows in the FS parameter (channels 10 to 20 and 60 to 80).

   $P1N/FL1/$P1S/DNA/$GATE/1/$G1N/FS/$R1I/1/$R1W/10,20/$R2I/1/$R2W/60,80/$GATING/R1.OR.R2/

   This sequence uses the **OR** operator so that if the FS signal falls in either window, the DNA data are recorded.

3. Store DNA vs RNA data for a polygon window in FS vs SS.

   $P1N/FL1/$P1S/DNA/$P2N/FL2/$P2S/$RNA/ $GATE/2/$G1N/FS/$G2N/SS/$R1I/1,2/$R1W/ 10,10;15,20;20,25;25,20;17,8;10,10/ $GATING/R1/

   The value of the **$R1W** keyword is the list of vertices of the gating polygon. The beginning vertex is repeated at the end of the list.

4. Store RNA data for events that have a FS signal falling between channels 75 and 125 and no SS signal falling between channels 25 and 50.

   $P1N/FL2/$P1S/RNA/$GATE/2/$G1N/FS/ $G2N/SS/$R1I/1/ $R1W/75,125/$R2I/2/$R2W/ 25,50/$GATING/R1 AND (NOT R2)/

   A gating signal must be present in region 1 (FS) but not in region 2 (SS) for the RNA signal to be stored.

APPENDIX A.

Major Differences Between FCS1.0 and FCS2.0

1. FCS2.0 has required keywords
2. A single data file may contain more than one data set. The **$NEXTDATA** keyword specifies the byte offset to additional data sets. For compatibility with FCS1.0, use **$NEXTDATA/0/**.
3. FCS2.0 contains the new keyword **$BYTEORD** to permit the reading of data files written by different computer operating systems. **$BYTEORD/1,2,3,4/** was the default in FCS1.0.

4. The **$ASC** keyword of FCS1.0 has been replaced by the **$DATATYPE** keyword in FCS2.0.

5. A "*" format has been added to FCS2.0 to allow the storage of ASCII-encoded data.

6. The **$PnE** and **$GmE** keywords have been added to specify the type of signal amplification used (linear or logarithmic).

7. FCS2.0 contains four new keywords (**$GATE, $RjI, $RjW, $GATING**) to provide for more flexibility in specifying gating conditions. The **$GmGnW** keywords are now obsolete.

8. The **$DFCmn** keyword has been changed to **$DFCiTOj**.

APPENDIX B. Data File Standards Committee of the Society for Analytical Cytology

Phillip N. Dean, Chairman	Biomedical Sciences Division Lawrence Livermore National Laboratory P.O. Box 5507 L-452 Livermore, California 94551
C. Bruce Bagwell	Verity Software House, Inc. Maine Cytometry Research Institute P.O. Box 247 Topsham, Maine 04086
Tore Lindmo	Biophysics Group Division of Physics University of Trondheim-NTH N7034 Trondheim-NTH Norway
Robert F. Murphy	Department of Biological Sciences Carnegie Mellon University 4400 Fifth Avenue Pittsburgh, Pennsylvania 15213
Gary C. Salzman	Biophysics Group Life Sciences Division Los Alamos National Laboratory Los Alamos, New Mexico 87545.