MRO FTT/NAS & FLC

The dlmsg FITS convention

MRO-MAN-CAM-1161-0166

John Young <jsy1001@cam.ac.uk>

rev 0.4

23 September 2014

Cavendish Laboratory
JJ Thomson Avenue
Cambridge CB3 0HE
UK

Change Record

Revision	Date	Author(s)	Changes
0.1	2014-08-08	JSY	Initial version based on "Network Message Protocols and Telemetry/Status/Logs File Format" rev 1.4
0.2	2014-08-29	JSY	Version for internal release
0.3	2014-09-05	JSY	Added Appendix A
0.4	2014-09-23	JSY	Changes to DATE keyword

Objective

To describe the FITS-based file format used by the Cambridge-supplied FTT/NAS and FLC control GUIs to record monitor data and to log diagnostic messages.

Scope

This document describes the FITS-based file format used by the `flcgui` and `fttgui` graphical user interface applications for recording status and telemetry and logging diagnostic messages. It is also used by the `testgui` application used in Cambridge for integrated stability testing of the FTT opto-mechanics. The format is a development of that used by the prototype delay line software, and we plan to use this updated format for the production delay line software.

Reference Documents

RD1 Definition of the Flexible Image Transport System (FITS), version 3.0, W. D. Pence, L. Chiapetti, C. G. Page, R. A. Shaw & E. Stobie, November 2010, Astronomy & Astrophysics 524, A42

RD2 A Hierarchical Grouping Convention for FITS, D. G. Jennings, W. D. Pence, M. Folk & B. M. Schlesinger, May 2007, `http://fits.gsfc.nasa.gov/registry/grouping/grouping.pdf`

Acronyms and Abbreviations

API Application Programming Interface
DL Delay lines
dlmsg Cavendish delay line messaging protocol
GUI Graphical User Interface
HDU Header-Data Unit
ISS Interferometer Supervisory System
MROI Magdalena Ridge Observatory Interferometer
URI Uniform Resource Identifier
URL Uniform Resource Locator

Table of Contents

1 Introduction

The Cambridge-supplied FTT/NAS and FLC GUI applications have the capability to record monitor data, log and fault messages (in ISS terminology) to a set of disk files. The disk files conform to the *convention* defined in this document, which itself conforms to the very flexible FITS standard [RD1]. The convention is based on FITS binary tables, as Matlab has a built-in capability to read these (they can also be read into C, Java, Python and IDL programs using third-party libraries), and the Cambridge team has relevant experience of writing FITS from C. The forthcoming production delay line software is expected to use the same convention for data recorded by its graphical user interface software.

Monitor data are transmitted continuously to the GUI application using the dlmsg network protocols. These data are inserted into a circular buffer on arrival, so that the most recent 100 seconds of data (the amount can be changed at compile time) are available most of the time. Log and fault messages are always written continuously to FITS files, but recording of other kinds of data are initiated by the user. All kinds of data are grouped into *sessions*, which are intended to group a related set of recordings (such as those from a sequence of tests) made during the same 24 hour period. A session consists of a directory containing multiple FITS-format files (this is for efficiency when writing the data). Within a session, data are grouped into recordings, each of which is initiated by the user and spans a continuous time period. Starting the GUI application automatically initializes a new session. The user may close the session and begin a new one by clicking a button on the GUI. The FITS files use the Heirarchical Grouping Convention (HGC) [RD2] to describe the relationship between the various FITS files in the session directory. In particular the recording application writes an index file which is subsequently read by the analysis GUI in order to discover the recordings belonging to the session and the files that comprise each recording.

The FITS convention used by the prototype delay line software has identical definitions of the constituent FITS binary tables, but the tables comprising a recording are stored in a common FITS file.

1.1 Virtual clients

In general, the data recorded by a control GUI instance originates from various components of the control software for a group of related systems. For historical reasons, these components are known as *virtual clients*. Each software process within the group of systems publishes data for one or more virtual clients. Each virtual client is identified by a short string, e.g. `FTTENV`, that is unique within the group of systems. Data messages contain an integer "ConfigId" that is incremented whenever the set of data items published by a virtual client or their sampling rates change (or publishing stopped and was subsequently restarted). When this occurs the circular buffers for that virtual client are destroyed and recreated. If a recording is in progress, writing to that client's existing FITS tables is stopped, and recording resumes using a new set of tables.

1.2 Telemetry & Status

The dlmsg protocols define two types of network message for the transmission of monitor data: "telemetry" messages and "status" messages. Both types are recorded by the control GUI using the convention described in this document.

The historical distinction between the two types of messages is that telemetry messages were designed for high sampling rate diagnostic data (e.g. from sensors) that need not be received with low latency, whereas status messages were designed for low sampling rate ($\lesssim$30 Hz), low latency data used in controlling the system. This distinction was blurred by the extension of the telemetry message format to accommodate image data, which must be displayed to the operator with reasonably low latency.

Telemetry messages were also designed to store precise timing information within a compact representation. A sequence of messages is sent by each virtual client. Each message within the sequence contains multiple data streams, identified as belonging to subsets with synchronous sampling. A message contains a contiguous set of data samples for each stream, called a “chunk”, as well as the timestamp of the first sample in the chunk and the precise sampling rate.

Status messages are used to transmit information needed by the software and/or the operator to control the system. The message content includes:

- Information (mostly boolean) about the state of a sub-system, which typically changes in response to commands.
- Information about whether the system is performing acceptably.
- Information that should be displayed in real time.
- Command acknowledgments, to provide near-immediate feedback on whether each command was accepted.
- Human-readable log/fault messages.

Status messages are sent at regular intervals (every $\sim$0.1 s). The message format allows these to contain arbitrary boolean and numerical status items.

Special components of each status message indicate whether any commands have been received since the previous status message was issued, and for each such command, whether the command and any associated parameters are valid, and whether the command will be acted on. In this way the status message incorporates command acknowledgement(s). Any subsequent changes of state in response to a command will be indicated by application-specific boolean status items.

Each status message may also contain any number of log/fault notifications, each comprising:

- a log/fault type specifying the category of log/fault and its level (severity),
- a bitmask (historically called a “trolley mask”) specifying which of the (up to 10) parallel systems the log/fault applies to, and
- a human-readable message.

In the case of faults, the first component of the message string is a name for the fault. The log/fault types implemented for status messages (see Table 1) have been chosen in order to map straightforwardly to the types and levels for logs and faults understood by the ISS. Log/fault messages are recorded to a different kind of table (Sec. 4.5) compared with the boolean and numerical status items (Sec. 4.4).

2 FITS Primer

FITS binary tables are part of the FITS standard (which is in widespread use in astronomy), and provide a framework (meta-format) for storing heterogeneous data in a compact binary form. The latest version of the FITS standard is version 3.0.

FITS files consist of one or more header/data units (HDUs), each of which represents an image, binary table, or ASCII table, together with associated metadata. Headers are always encoded in ASCII, and contain a set of keywords and associated values. Certain keywords have special meanings according to the FITS standard (for example they describe the structure of the data part of the HDU), but other application-specific keywords can be included. For historical reasons, the first HDU can only contain an image (which may be of zero size).

2.1 Hierarchical Grouping Convention

The dlmsg FITS convention makes use of the Hierarchical Grouping Convention (HGC) [RD2], which is a registered FITS convention – see `http://fits.gsfc.nasa.gov/fits_registry.html` for an up-to-date list. The HGC is supported by the CFITSIO software library and the FV FITS reader.

A group table, as defined by the HGC, is a FITS table extension that contains a list of all the associated member HDUs in the group. Group tables may be represented by either FITS ASCII tables or binary tables. There may be zero, one, or more group tables within a given FITS file. Each group table may reference any number of HDUs. The entire set of HDUs referenced in a group table, along with the group table itself, form a group. Individual HDUs referenced in a group table are said to be members of the group or group members. Groups can contain any type and mix of HDU. Note that a group may also contain other groups as members. This feature allows for the construction of hierarchical structures of HDUs within a single FITS file or across many FITS files.

The HGC strongly recommends that HDUs belonging to one or more groups contain keywords that "point back" to the location of the relevant group tables. The GRPID*n* keyword gives the integer value of the EXTVER keyword for the nth group table to which the HDU belongs. A negative value for GRPID*n* indicates that the group table resides in another file (and removing the negative sign gives the group table's EXTVER). In this case an additional GRPLC*n* keyword must be present, to specify the URI of the file containing the group table.

3 Data groupings

Each HDU containing monitor data (DL_STATUS or DL_TELEMETRY) shall be stored in a separate file. This is for efficiency reasons, to avoid having to reserve space in each HDU at the start of recording, and remove unused reserved space if recording is aborted. The HGC must be used to group contemporaneous monitor data HDUs into a "recording" group. Recording groups must themselves be grouped into "session" groups.

The grouping HDUs defining both a session group and the recording groups it contains must be stored in a single FITS file, named "index.fits". These HDUs must contain the start and end times of the session and recordings and the names of the virtual clients that were recorded, as detailed in Sec. 4.2.

Log messages must be stored in a single DL_LOG HDU written to a separate file, named "log.fits" and belonging to the session group only. This is because logs are normally recorded continuously while the GUI is running, rather than for short timespans initiated by the user.

Reading software must not rely on anything else about the filenames used.

4 FITS tables

This section defines the content of each of the HDU types in the dlsmg FITS convention.

Where the repeat count in a TFORMn specification below is unimportant, it is represented by an asterisk. For example "*A" could correspond to "8A" or "16A" in an actual file.

Reading software should not assume anything about the ordering of tables within a file, or keywords or columns within a HDU.

4.1 Primary HDU

The FITS standard mandates that the first HDU in a FITS file must be an image HDU. This HDU is not used in the dlmsg convention, and the image should therefore be of zero size.

4.2 GROUPING table

This section defines a GROUPING table as used in the dlmsg FITS convention. This usage is consistent with the HGC specification [RD2]. Such tables must be used to define a session group and zero, one or more recording groups that are members of the session group.

Under the dlmsg convention, group members are always binary table HDUs. The DATE-END keyword and the CLID column are specific to the dlmsg convention.

Keywords defined by the FITS standard

Keyword	**Type**	**Value**	**Example**
EXTNAME	character	`GROUPING`	`GROUPING`
EXTVER	integer	Unique group ID within file	`1`
DATE-OBS	character	Start UTC of session/recording as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-08-28T10:48:30.053`
DATE	character	UTC when HDU written as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:09:37`

Other keywords defined by the FITS standard describe the structure of the binary table and are normally handled transparently by FITS readers; these include NAXIS, NAXIS*n*, TFIELDS, TTYPE*n*, TFORM*n*, and TUNIT*n*.

Other keywords

Keyword	**Type**	**Value**	**Example**
GRPNAME	character	Grouping table name	`REC01`
GRPID1	integer	(optional) ID (EXTVER) of 1st parent group (recording group has a session group parent)	`1`
DATE-END	character	End UTC of session/recording as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-08-28T10:49:30.012`

Columns

Column	Type	Value	Example
CLID	*A	(optional) Client identifier (omit if session group)	`FTTENV`
MEMBER_XTENSION	*A	XTENSION of member HDU	`BINTABLE`
MEMBER_NAME	*A	EXTNAME of member HDU	`DL_STATUS`
MEMBER_VERSION	1J	EXTVER of member HDU	`1`
MEMBER_POSITION	1J	Position of member HDU in file (primary array has position 1[1])	`2`
MEMBER_LOCATION	*A	(optional) Location (URI) of member's FITS file (`NULL` if member in same file)	`REC01_20140618_140937_FTT_001_status.fits`
MEMBER_URI_TYPE	*A	(optional) Type of URI i.e. `URL` or `URN` (`NULL` if member in same file)	`URL`

The HGC specifies that the MEMBER_LOCATION and MEMBER_URI_TYPE columns may be omitted if all group members reside in the same file as the GROUPING HDU. This is the case for dlmsg session groups, which may therefore omit these columns, but not for recording groups.

4.3 DL_TELEMETRY table

The telemetry streams in each table are those that are time-synchronised with each other (as indicated by the secondary client identifiers in the telemetry messages), and thus there will be at least one table per active virtual client for each time interval.

Keywords defined by the FITS standard

Keyword	Type	Value	Example
EXTNAME	character	`DL_TELEMETRY`	`DL_TELEMETRY`
DATE-OBS	character	Start UTC of table data as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:09:36.980`
DATE	character	UTC when HDU written as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:09:37`
TTYPE*n*	character	Name of column (field) *n* (=stream identifier or `UTC`)	`FRAME`
TUNIT*n*	character	Units for column *n*	`dn`
TDIM*n*	character	(optional) Dimensions for column *n*	`(32,32)`

Other keywords defined by the FITS standard describe the structure of the binary table and are normally handled transparently by FITS readers; these include NAXIS, NAXIS*n*, TFIELDS, and TFORM*n*.

Other keywords

The GRPID*n* and GRPLC*n* keywords are defined by the HGC [RD2].

[1] The HGC specifies that the HDU numbering used with MEMBER_POSITION starts at 1 for the primary array, but CFITSIO and FV behave as if the numbering starts at 0.

Keyword	Type	Value	Example
TBL_VER	character	Version number of table definition	`1`
CLID	character	Client identifier for streams in this table	`FTT-RUN`
SEC_CLID	integer	(optional) Secondary client identifier for streams in this table	`1`
REFSTRM	integer	Number of column containing reference stream data	`3`
SMPRATE*n*	floating point	Sample rate for column *n* /Hz	`248.447196`
TIMOFF*n*	integer	(optional) Time offset from reference stream for column *n* /μs	`23`
DATE-NOM	character	Start UTC of recording as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:09:37.028`
UTC-NOM	floating point	Start UTC of recording as Unix Time	`1403100577.02819`
GRPID1	integer	ID (EXTVER) of recording group (negative $\implies$ external file)	`-2`
GRPLC1	character	File for recording group	`index.fits`

Columns

A DL_TELEMETRY table must contain a column named UTC containing timestamps, plus one or more columns containing regularly sampled data.

Time series of floating point, integer, or boolean data may be stored in the table. Each data sample may be multi-dimensional, for example an image. The table's set of data streams are assumed to be sampled synchronously, with sampling rates that are different multiples of a base sampling rate. As mentioned above, a sequence of telemetry messages from a virtual client may contain multiple sets of synchronous data streams; each set must be written to a separate DL_TELEMETRY table.

The telemetry messages from which the stored data are copied contain just one timestamp for every "chunk" of each stream's time series, the others being implicit (the sampling rate is included in the message header). For example, a 5 kHz sampled stream might be transmitted in chunks of 5000 samples, each accompanied by one timestamp (that of the first sample in the chunk). A DL_TELEMETRY table must only contain explicit timestamps copied directly from telemetry messages.

The amount of storage space needed for timestamps is further reduced by only storing timestamps for *one* of the synchronous data streams, the so-called *reference stream*. Timestamps for the reference stream shall be stored in the UTC column, expressed in Unix Time (i.e. seconds since midnight Jan 1, 1970) and encoded as double precision floating point values. The reference stream shall be identified by the REFSTRM keyword. This must correspond to the most rapidly-sampled stream in the table. If several streams have equal-fastest sampling, one must be chosen arbitrarily as the reference stream to which the timestamps apply.

When reading the table, the UTC column and the SMPRATE*n* and TIMOFF*n* keywords shall be used to infer timestamps for all of the data streams.

The table should normally be structured such that each row corresponds to a single chunk of telemetry for the reference stream (although it would be legitimate to store multiple reference stream chunks in each row). For two synchronised streams A and B with sampling rates of 5 kHz and 10 Hz respectively, the standard table structure would be as follows (where (. . .) represents a single cell):

(1×UTC) (5000×A) (10×B)

(1×UTC) (5000×A) (10×B)
(1×UTC) (5000×A) (10×B)
etc.

In general, each binary table "cell" contains a multi-dimensional *array* of telemetry data. If there is more than one dimension, the cell dimensions must be specified with the standard FITS TDIM*n* keyword. The slowest (last) axis shall be the time axis, with additional dimensions corresponding to the dimensions of a data sample. For scalar data there would be a single dimension, time. For greyscale image data, there would be three dimensions: x, y, and time. In a future application colour images could be stored using dimensions x, y, RGB, and time. Usually one image is written per table cell[2], in which case the time dimension may be omitted from TDIM*n*.

The display orientation for the images shall follow the FITS convention i.e. the image pixel (1,1) is displayed in the lower left corner, the X coordinate increases to the right in the image, and the Y coordinate increases in the upward direction.

The time (slowest) dimensions for different columns must be chosen such that the data in each row of the table spans the same time interval for all columns. The data type of each column shall match that used in the telemetry messages for that stream. Possible data types are FITS logical (boolean), integer, and floating point types.

4.4 DL_STATUS Table

The data written to this table from each status message comprise one or more groups (message units), each consisting of a set of boolean items and a set of floating point items plus an accompanying timestamp. Multiple groups are used to encode the measurement times more precisely. All of the groups sent by a particular client are written to a common table.

Keywords defined by the FITS standard

Keyword	Type	Value	Example
EXTNAME	character	`DL_STATUS`	`DL_STATUS`
DATE-OBS	character	Start UTC of measurement as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:09:36.980`
DATE	character	UTC when HDU written as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:09:37`
TTYPE*n*	character	Name of column (field) *n* (=status item label or `UTC`)	`KALMANBANDWIDTH`
TUNIT*n*	character	Units for column *n*	`Hz`

Other keywords defined by the FITS standard describe the structure of the binary table and are normally handled transparently by FITS readers; these include NAXIS, NAXIS*n*, TFIELDS, and TFORM*n*.

Other keywords

The GRPID*n* and GRPLC*n* keywords are defined by the HGC [RD2].

[2]The current implementation only supports writing one image per table cell.

Keyword	Type	Value	Example
TBL_VER	character	Version number of table definition	`1`
CLID	character	Client identifier for status items in this table	`FTT`
DATE-NOM	character	Start UTC of recording as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:09:37.028`
UTC-NOM	floating point	Start UTC of recording as Unix Time	`1403100577.02819`
GRPID1	integer	ID (EXTVER) of recording group (negative $\Longrightarrow$ external file)	`-2`
GRPLC1	character	File for recording group	`index.fits`

Columns

A DL_STATUS table must contain a column named UTC containing timestamps, plus one or more columns containing boolean or floating point data. There must be one value per table cell.

The table shall have columns of FITS logical type with a single boolean status value per table cell (`TFORMn = '1L      '`), and columns of double precision type with a single numeric status value per table cell (`TFORMn = '1D      '`). The timestamps (Unix Time) for the status information must be stored in a double precision column with `TTYPEn = 'UTC     '`.

If a client groups its status items into more than one message unit, one FITS row shall be written per unit. In each row, NULL values shall be written to the columns for status items belonging to other message units. NULL values shall be encoded as per the FITS standard [RD1] i.e. zero-valued byte for logical columns, IEEE NULL for double precision columns.

Log/fault messages are written to a separate FITS table (Sec. 4.5).

Command acknowledgements must be stored in the mandatory columns listed below. If the number of acknowledgements in a status message exceeds the number of message units (and hence the number of FITS rows written per message), an extra table row shall be included for each further command. Besides the acknowledgement columns, other columns in these additional rows shall contain the same values (including timestamp) repeated.

Column	Type	Value	Example
ICMD	1I	Index of command in interval since last status message	`1`
CMDSRC	*A	Source of command	`SUPERVISOR`
CMDTAG	1I	Command tag	`32`
PFLAGS	3L	Parse flags	`TTF`

If there is no command acknowledgement (because no command was received since sending the previous status message) a negative value shall be written for ICMD. In this case CMDSRC, CMDTAG and PFLAGS must be ignored when reading the file.

4.5 DL_LOG Table

This table is used to record human-readable log and fault messages received from clients in status messages. The various categories of log/fault messages are enumerated in Table 1.

Table 1: Log/fault type codes used in status messages. Note that "SEVERE FAULT" has been removed for consistency with the MROI Generic System Interface framework.

Value	Type	Maps to MROI LogType	MROI LogLevel	Comment
1	VERBOSE	DEBUG	FINER	
2	DEBUG	DEBUG	FINE	
3	CONFIG	CONFIG	CONFIG	Related to configuration, especially at startup
4	INFO	INFO	INFO	
5	EXECUTED	INFO	INFO	Message contains dlmsg command tag
6	WARNING	WARNING	WARNING	
7	FAULT	FAULT	WARNING	
8	EXCEPTION (CLIENT)	EXCEPTION	SEVERE	Message contains command tag
9	EXCEPTION (INTERNAL)	EXCEPTION	SEVERE	Message optionally contains command tag

Keywords defined by the FITS standard

Keyword	**Type**	**Value**	**Example**
EXTNAME	character	`DL_LOG`	`DL_LOG`
DATE-OBS	character	Start UTC of messages as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:02:07.167`
DATE	character	UTC when HDU written as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:02:07`

Other keywords defined by the FITS standard describe the structure of the binary table and are normally handled transparently by FITS readers; these include NAXIS, NAXIS*n*, TFIELDS, TTYPE*n*, TFORM*n*, and TUNIT*n*.

Other keywords

The GRPID*n* and GRPLC*n* keywords are defined by the HGC [RD2].

Keyword	**Type**	**Value**	
TBL_VER	character	Version number of table definition	`1`
GRPID1	integer	ID (EXTVER) of recording group (negative $\Longrightarrow$ external file)	`-1`
GRPLC1	character	File for recording group	`index.fits`
DATE-END	character	End UTC for session as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:09:53.843`

Columns

Column	Type	Value	Example
UTC	1D	Timestamp (Unix Time)	`1403100577.02819`
CLID	*A	Source of message	`FTTENV`
TYPE	*A	Category of log/fault message (see Table 1)	`FAULT`
TRLYMASK	10L	System mask, `T` for affected systems	`TFFFFFFFFF`
TIME-OBS	*A	UTC timestamp as *hh:mm:ss[.sss]*	`14:09:37.028`
MESSAGE	*A	Human-readable message. For FAULT and SEVERE FAULT types, the message begins with the fault name followed by a colon.	`EnclosureTooHot: enclosure is too hot`

A Dark/sky/flatfield image convention

The dark/sky and flatfield images captured by the FTT system and transmitted as dlmsg telemetry are saved by the control GUI, each image being written to a separate FITS file. The file shall contain a two-dimensional image in the primary HDU. The keywords listed below shall be written to the primary header.

Keywords defined by the FITS standard

Keyword	Type	Value	Example
DATE-OBS	character	UTC when image taken *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:02:07`
DATE	character	UTC when HDU written as *yyyy-mm-dd*T*hh:mm:ss[.sss]*	`2014-06-18T14:02:07`

Other keywords defined by the FITS standard describe the structure of the image data and are normally handled transparently by FITS readers; these include NAXIS, NAXIS*n* and BITPIX.

Other keywords

Keyword	Type	Value	Example
NFRAMES	integer	Number of frames averaged	`100`
EXPOSURE	floating point	Exposure time of each frame /s	`0.1`
SHUTOPEN	logical	Was the shutter open for the exposure?	`T`
CUSTOM	logical	Was custom readout mode used?	`F`
SUBRECTX	integer	(optional) X coordinate of CCD readout region /pix	`240`
SUBRECTY	integer	(optional) Y coordinate of CCD readout region /pix	`240`