

Pointing and live time history

 [Jump to the updates of 26 Sep 2008](#)
 [Jump to the updates of 19 Aug 2008](#)
 [Jump to the updates of 20 Nov 2007](#)

 [New implementation! \(Jan. 2012\)](#)

Pointing and live time history (FT2, LS-005)

This is a fundamental data product used by the science tools. The current definition ([version 10 of July 25, 2006 as of this writing](#)) is posted on the [Guidelines for Science Tools Design](#) page maintained by Masa Hirayama. This page includes a link to the ft2.tpl template file that is the current working definition of the FT2 format as used by the Science Tools.

Basically, the file is intended to include the necessary information (other than tables of effective area) for calculating exposures. So it has the position and attitude of the LAT for regularly-spaced time intervals (nominally 30 seconds, although shorter intervals are used when the LAT enters and exits the South Atlantic Anomaly). As currently planned, SAA entry and exit will correspond to the stop and start time for the data taking 'runs' in flight. For orbits that miss the SAA, a run stop/start will be issued when the LAT crosses the orbital plane heading north. [anders] The reasons for the forced stop/starts can be found [here](#). For these stop/starts, the interval between the entries may also be less than 30 s. For each of the intervals in an FT2 file the accumulated live time is also recorded.

This page is open for comments, and for posting additional issues. For clarity, please add the comments in the section to which they refer, by editing the page directly. When the comments settle down, I will post as JIRA issues any specific changes that we converge on for the definition of the FT2 format.

1. Interval between entries in the FT2 file

(Dige) The 30-second interval was chosen as being relatively long but short enough (corresponding to ~2 deg of advance of the pointing position during scanning observations) so that the exposure could be calculated accurately. The thinking was that 2 deg is so much smaller than the FOV of the LAT that accurate exposures could be calculate with 30-second intervals.

Three questions:

GLAST will slew much more quickly when it is rocking - perhaps 15 deg per minute (?) - I think the spec is 75 deg in 10 minutes but GLAST is said to be significantly faster. Do we care in terms of the time spacing of the FT2 entries? The science tools do not assume any particular spacing, so in principle a shorter interval could be used when the LAT is rocking. Is it worth worrying about?

And when the LAT is just scanning, can we make do with a longer interval, say 1 minute? How severe is the tradeoff in accuracy? The motivation for making the interval longer would be to make FT2 files smaller and to increase the speed of exposure calculations.

This is kind of a minor point, but may need to be clarified: During the time between the stop (LPASTOP) of a run and the LPASTART of the next (if a stop/start is issued outside an SAA passage - see above), the LAT will be dead as far as data taking is concerned. If a time interval is not explicitly included in an FT2 file, i.e., if the time of a START entry is not the same as the time of the previous STOP entry, we can just assume that the missing interval was dead time, right? [anders] Any time between an LPASTOP and the subsequent LPASTART is deadtime i.e. we did not take any data.

(D. Band 4/6/07) First, is this an ICD/FFD issue, given that every row provides the start and stop time? Second, I think it makes sense to increase the time resolution when necessary--perhaps we should establish an angular change criterion?

(J. Chiang 4/23/07) We need to establish a maximum time interval size to facilitate the reading of the FT2 files by our various tools. For GRB analysis, the burst duration may be much less than 30 seconds (or whatever the nominal time interval is) and if one uses the extended filename syntax to filter on FT2 data using the inferred burst time boundaries, (tmin, tmax), using a filter string "(START >= tstart) && (STOP <= tstop)", and if those boundaries lie entirely within a time interval, this filter string will return zero intervals. To guard against this we would need to do something like "(START >= tstart-maxIntervalSize) && (STOP <= tstop+maxIntervalSize)".

2. DEADTIME column

(Dige) For reasons that I no longer recall, we originally decided to keep track of both accumulated livetime (in each 30 second interval) and accumulated deadtime (since the start of the mission). The Science Tools do not use the DEADTIME column and **makeFT2** assigns the value 1-LIVETIME to the column anyway.

I propose to remove this column from the definition of FT2.

(D. Band 4/6/07) I concur with removing this column.

(Ormes, from an e-mail message 4/14/07) The deadtime comes naturally from properties of the electronics and is normally easy to track, but as you have assumed in making the above decision, not what we want. We need live time to compute fluxes. Live time cannot reliably be obtained by subtracting dead time from clock time, no matter how much attention is paid to removing time in SAA, spacecraft anomalies, etc. I pushed for a requirement that the electronics contain a clock that measures the livetime between events. One should be able, from this, to plot an "interval distribution" which should be exponential, with a mean that reflects the inverse of the counting rate. If the live time distribution, for any given type of event is not exponential, there is trouble.

However, this said, if there is trouble, we should have the deadtime, too, for cross checks. It should be filled with the number of events times the deadtime per event, or if the deadtime per event is not exactly constant but varies with the amount of readout or something else, the sum of the deadtimes over the events obtained during the accumulation time in question. I recommend not dropping the deadtime column but filling it with something useful.

(Digel, from reply 4/15/07) I'm not sure what you mean by trouble, whether if the interval distribution does not turn out to be exponential (because of a hardware problem) we'll be able to do anything about it. As far as I know, even if we filled the DEADTIME column with the number of events times the average deadtime per event in each interval, the average deadtime would be derived from the livetime counter values. So I'm not sure that the DEADTIME column would give us any independent information that would be useful for correcting live times.

We'll certainly want to make sanity checks on the livetimes vs. the numbers of triggers, but I think that we don't need to carry the DEADTIME column in the FT2 file. We'll either decide that everything is ok with the LIVETIME determinations or that there's a problem. If there's a problem, I think we won't distribute FT2 files until it is solved, rather than make FT2 files for the user in the street to potentially misinterpret.

(Ormes, from e-mail 4/16/07) I don't know how the electronics was finally done. I do know these numbers, live and dead were supposed to be measured independently in the electronics. Maybe it was not done in the end.

(Borgland, from e-mail 4/16/07) There is no deadtime counter since by definition the instrument is dead. There is only a livetime counter which increments in 50 ns ticks all the time the instrument is alive i.e. not busy/dead reading out etc. There is also an elapsed counter which is the total time. The livetime fraction between events N and N+M is just
$$\text{LiveTimeFraction} = \frac{\text{LiveTime}(N+M) - \text{LiveTime}(N)}{\text{ElapsedTime}(N+M) - \text{ElapsedTime}(N)}$$

and the deadtime (for whoever is interested in that) is just $1 - \text{LiveTimeFraction}$.

Concerning this statement:

However, this said, if there is trouble, we should have the deadtime, too, for cross checks. It should be filled with the number of events times the deadtime per event, of if the deadtime per event is not exactly constant but varies with the amount of readout or something else, the sum of the deadtimes over the events obtained during the accumulation time in question.

This is not possible because of the onboard filter. We do not know what trigger engine caused the readout of events that were rejected by the filter i.e. how they were read out (4-range, non zero-supp etc). Note that I actually currently do what Jonathan describes here in the current digi report. However, it's only valid for runs not running the filter (or running the filter in the passthru mode).

(Ormes, note added 4/20/07) - The implementation meets my original intentions. Having the elapsed total time and the live time are sufficient for the necessary cross checks.

5/16/07: Removing the DEADTIME column has been submitted as a JIRA request, <https://jira.slac.stanford.edu/browse/DATAPROD-1>

3. McIlwain coordinates

(Digel) These are geomagnetic coordinates and presumably useful in some way for studying variations in residual background. The actual model we are using for the background depends on geomagnetic latitude, however, rather than the McIlwain L, B. I'm not entirely sure that we need (i.e., that anyone will use) L, B, but I won't propose removing them. I would like to propose including geomagnetic latitude, however, as an additional column.

(D. Band 4/6/07) I concur with adding this column (especially since I don't have to do the calculation...).

(Ormes 4/20/07) Thanks. This will be helpful and avoid having any confusion that might be caused by the user having a separate lookup table to get it back.

5/16/07: Adding the geomagnetic latitude has been submitted as a JIRA request, <https://jira.slac.stanford.edu/browse/DATAPROD-2>

4. IN_SAA column

(Digel) This is just a flag defined to be True if the LAT is in the SAA. Right now it is not entirely clear (to me) whether we will be getting spacecraft position and attitude information during SAA passages. This would have implications, e.g., for using the Low-Rate Science counters to monitor the boundaries of the SAA, and also would mean that an IN_SAA flag would be superfluous. I'll update (or remove) this entry when I find out the answer.

(Digel) Here's word from Bryson Lee (4 April 2007): "The magic-7 data are sent out in the science data stream whenever the SIU is in application mode (after secondary boot) and the LIMMAINFEEDON command has been issued to power up the PDU and GASU. In effect, the data are available whenever the LAT is powered on, regardless of whether or not any events are being acquired."

So the IN_SAA flag remains relevant to the definition of FT2.

(D. Band 4/6/07) I understand this to mean that we will keep this column.

5. Extension Name for data extension in FITS file

(Stephens) I don't know if this has ever been discussed before but I noticed a discrepancy between the data and the documentation and thought I'd bring it up. We should verify that the science tools that both produce and consume this product are using the correct extension name (EXTNAME keyword value) for the extension containing the data and that that name matches what is in the documentation. The Science Data Products ICD and File Format documents list the extension name for this extension as 'LAT_POINTING_HIST'. It formerly used to be 'SC_DATA' and the last time I checked (mid-late January) the Science Tools were still set up to use this older name. While the name doesn't really matter, what does matter is that we are consistent across the tools and the documentation. The GSSC's data ingest system works off the ICD and that document is what the HEASARC will look to as well for the data formats so we need to keep it up to date and in sync with the software. Because of our requirements on data validation and tracking, if data arrives at the GSSC that doesn't match the format in those documents, the data will be rejected as bad. The bottom line is that we should settle on a name for this extension, have David Band put it in the document, make sure all the software are using that name, and be done with it.

(D. Band 4/6/07) I am not sure which name is older. I have not strong feelings one way or the other.

(*J. Chiang 4/23/07*) Unless there is a good reason to change it to 'LAT_POINTING_HIST', I recommend that we change the documentation to use 'SC_DATA'. I recall having exchanges/conversations agreeing to change it from the obscure 'Ext1' to 'SC_DATA'. I do not recall similar exchanges regarding any change to 'LAT_POINTING_HIST'.

5/16/07: We'll stick with SC_DATA. This means that the Science Data Products File Format Document will need to be updated.

Here is an additional item added on 20 Nov 3007

6. Attitude quaternions in FT2?

(*Dige!*) In the FT2 file the attitude of the spacecraft is expressed via the directions (in RA, Dec) of the z and x axes. Gleam and the astro package in general use quaternions to represent the attitude, and the 'Magic 7' information from the spacecraft does express the attitude via a quaternion. Some users of the Science Tools are advocating that FT2 be extended to include the quaternions in addition to the RA, Dec of the x and z-axes.

This is an issue primarily of convenience, although Toby has noted that precision errors can decrease the accuracy of the attitude when the x and z-axis directions are derived from a quaternion. The errors in precision are small and might be most relevant to the case where Gleam is driven by an ASCII equivalent of a pointing history.

In terms of the Science Tools, I don't think that anything is fundamentally broken with how we are expressing the attitudes in FT2 files, and perhaps too often we consider FT2 files as input files for Gleam (to specify the attitudes) rather than as a pointing/lifetime history useful for calculating exposures. That said, I don't have any objection to including the quaternion in the FT2 files.

Would single precision floating point provide adequate resolution?

I don't see any downside to including attitude quaternions, other than increased sizes for FT2 file, but I'm likewise not sure whether I see any big advantages either.

Here are additional items added on 19 August 2008

7. Rocking angle of the LAT

(*Dige!*) The rocking angle of the LAT (i.e., the direction of the LAT z axis from the zenith) is not used directly by the Science Tools but it is of interest for evaluating the observing strategy and especially for assessing albedo backgrounds and loss of exposure for pointed observations. It can be derived from the direction of the LAT z axis and the zenith direction, but having it directly in the FT2 file would be more convenient.

9/26/08: We'll add this as ROCK_ANGLE. One commenter requested that this be a signed quantity - positive for rocking toward the north orbital pole - that is what we'll do.

8. Direction of the orbital pole

(*Dige!*) Similarly, this information is not used in the Science Tools but having it conveniently available would be useful for evaluating observing strategies. The direction of the orbital pole can be derived from information already in the FT2 file but it requires, e.g., calculating cross products of zenith directions in successive time steps. Having the direction of, say, the northern orbital pole directly available in the file would be a convenience; the impact would be the addition of 2 floating-point columns. Also `ft2Util` would need to be updated. No existing FT2 file would 'break' as the result of this change.

9/26/08: We'll add this as RA_NPOLE, DEC_NPOLE. This is not needed by the Science Tools but is very convenient for browsing the pointing history, e.g., for selecting time ranges when a given direction was best covered by the LAT.

Here are additional items added on 26 Sep 2008 to summarize discussions that have taken place mostly via e-mail so far regarding needed updates for dealing with flight data.

9. LAT configuration flag

(*Dige!*) This flag abstracts the MOOT key and filter selection settings to indicate whether the data can be considered the equivalent of 'nominal Science Operations' quality for the Science Tools. This ordinarily would apply to a whole run but needs to be a column in FT2 because in various data servers the distinctions between runs will be lost. Right now the quantities envisioned for this flag are 1 = nomSciOps or close enough; 0 = not recommended for analysis in Science Tools

9/26/08: We'll add this as LAT_CONFIG

10. LAT data quality flag

(*Dige!*) This flag is the equivalent of the (per-run) data quality flags. It is defined on a per-run basis, but it needs to be a column in FT2 because in various data servers the distinctions between runs will be lost. Values currently envisioned are 1 = ok, 2 = waiting review, 3 = good with bad parts, 0 = bad

9/26/08: We'll add this as DATA_QUAL

See [Draft FT2 Template 26 Sep 2008](#) for specifics