



## Timing Analysis with Lightcurves

Return to: [Why Index](#)

This document discusses caveats that one should be aware of when doing timing analysis on Chandra data. The [CIAO Timing Threads](#) – primarily the [Basic Lightcurves](#) thread – show how to create and use lightcurves in your analysis.

### Tool: dmextract vs. lightcurve

As of CIAO 3.0, the `lightcurve` tool will no longer be actively maintained or enhanced. Users instead are encouraged to create lightcurves with the new functionality of `dmextract`, which more carefully and accurately applies good time interval (GTI) information. More information is available from the help files for `dmextract` and `lightcurve`

This `lightcurve` command:

```
unix% lightcurve infile="acis_evt2.fits[ccd_id=3,sky=region(src.reg)]" \
  outfile="lightcurve.fits" binlength="259.28"
```

becomes the following for `dmextract`:

```
unix% dmextract \
  infile="acis_evt2.fits[ccd_id=3,sky=region(src.reg)][bin time=:259.28]" \
  outfile="lightcurve.fits" opt="ltcl"
```

### Barycenter Correction

In general, a barycenter correction should be applied to any data before it is used for timing analysis. Barycentering adjusts the times associated with the observation (both the times associated with the events and the times in the header keywords) for the difference in photon arrival times as the Earth and Chandra move around the Sun. For more information on barycenter–correcting times, see the [Apply Barycenter Correction](#) thread.

### ACIS Lightcurves

The simplest lightcurve we can create is from a point source in ACIS data. This is generally done to get an idea of the variability of the source, or to look for background flares that should be filtered out.

#### *Caveats:*

- *False Periods, Aliasing*

The most important caveat in the creation of lightcurves is to caution against bin lengths near or less than the absolute temporal resolution of the data. For example, most ACIS data is taken in Timed Exposure (TE) mode, collecting photons and reading them out periodically (every ~3.2 sec, by default; stored as the `TIMDEL` keyword in the event file header). Each readout is called a "frame"

## Timing Analysis with Lightcurves – CIAO 3.4

and every event in a given frame has the same time assigned to it. If one attempts to create a lightcurve with a resolution similar or less than the readout time, the lightcurve will have several bins with zero count rate and false periodicities.

Bin sizes larger than the natural binning can still have aliasing problems. In principal, any lightcurve binned with a non-integer multiple of the intrinsic binning will show a false harmonic period with length equal to the difference of the bin size and the readout period. In practice, this affect is diminished when the periods are very dissimilar. If the bin length must be of order the readout period, set it equal to an integer multiple of the intrinsic bin size.

Another instrumental period may be seen if a source dithers across a bad pixel. The exposure information is not taken into account when generating the count rates, so the count rate will dip with a period equal to the dither rate as a source moves across a bad pixel. To check if your source is near a bad pixel region, view the level 2 event file in detector coordinates, as demonstrated in Displaying an Event File in Different Coordinates section of the ds9 thread.

In general, you should be cautious of any period which is similar to the dither period or a harmonic of the dither period and the lightcurve bin size (see this thread on the Chandra Users' discussion list for an example of this issue). This gets complicated quickly since the dither velocity is not constant and the effect will change depending on the alignment of the source and bad pixel(s). The nominal dither periods are:

	Peak-to-peak Span (arcsec)	Nominal Period in Y (s)	Nominal Period in Z (s)	<u>POG Reference</u>
ACIS	16	1000	707	<u>ACIS chapter</u>
HRC	40	1087	768	<u>HRC chapter</u>

Further details on the dither may be found in the Dither why topic.

### • **Multi-chip Timing Analysis**

Normally, the count rate is taken to be counts per bin per total observational good time. When `dmextract` is run with `opt=ltc1`, however, the count rate is actually counts per time bin, merged with the intersecting good time intervals, taken from the first GTI in the case of multiple GTIs (e.g., as for regions that span multiple ACIS chips). On account of this, some caution is necessary when dealing with multi-chip timing analysis.

GTIs are created on a per-chip basis because each chip functions as an individual detector to a certain extent. The GTI for each chip of an observation will be identical in most cases, but telemetry saturation may cause some chips to lose data. This will lead to differences between the GTIs of chips for ~20% of observations. Blindly creating lightcurves from these observations will generate incorrect results because the tool assumes that the aimpoint GTI is valid for the whole observation. Note that this problem will also arise if time filters have been applied to a subset of the chips (e.g. to filter the back-illuminated chips for background flaring).

The solution is to add a `ccd_id` filter when extracting your lightcurve, as shown in the "Tool: `dmextract` vs. `lightcurve`" section, above. This technique obviously will not work for multi-chip sources, which require special analysis regardless (i.e. the application of a single GTI is the least of the problems). For multi-chip sources with non-negligible GTI differences, the chips should be analyzed individually; check the ONTIME header keywords to find the exposure time for each chip in the observation.

## HRC Lightcurves

The proper method of creating an HRC lightcurve requires accounting for the Dead Time Factor (DTF), as illustrated in the [Basic Lightcurves](#) thread.

### *Caveats:*

- ***HRC–I Timing Accuracy***

After launch, the HRC–I detector was found to have a hardware timing error which reduced timing accuracy from ~16  $\mu$ sec to ~4 msec. The problem is described in the [HRC timing caveat](#).

## Timing with ACIS CC–Mode Data

The [ACIS continuous clocking mode](#) may also be used for timing analysis. Observing in this mode produces a 1–D "image" of the sky by continually reading out the detector at a rate of 2.85 msec/row.

### *Caveats:*

- ***Coordinates***

Since the value of CHIPY for an event is unknown, the sky coordinates are calculated assuming that the target–row events (those events with CHIPX expected for the target) have the target RA and Dec. Note that this is the only location in the observation which has meaningful (RA,Dec) coords. All other events are chosen to have the same CHIPY value for aesthetic reasons. The source–pixel assumption will make timing analysis of extended or crowded sources difficult, since the arrival time of the non–target events will be off by

$$(\text{CHIPY\_actual} - \text{CHIPY\_target}) * 3 \text{ msec/row.}$$

- ***Event Detection***

As described in the [POG](#), the event detection algorithm looks for local maxima in 3x3 pixel event islands in a 1024x512 pixel "virtual detector" in the CCD memory buffer. This detection method cannot find events at the border of the buffered image, meaning that no event will ever be found with CHIPX of 1 or 1024 and CHIPY of 1 or 512. The CHIPY limitation is important for timing analysis, since it means that events cannot occur at certain times separated by  $512 * 2.85$  msec or 1.4592 sec. For the same reason, it is impossible for two events to occur in the same column in adjacent time bins.

- ***Times of Arrival***

The times assigned to events in CC–mode data are ***readout times***, not times of arrival. Absolute timing analysis requires that the data be corrected before creating lightcurves. Information on how to do so is available in the [Calculate CC–mode Times of Arrival](#) thread.

## Timing with an ACIS Readout Streak

Bright sources observed by ACIS are typically problematic in data reduction, but the "readout streak" may be considered an exception. As it does not have a shutter, ACIS continues to detect photons while reading out a frame (which takes ~41 msec/frame). During this short time, it is probable that photons from bright sources (e.g. a 1 count/s point source) will strike the detector. These events will be read with CHIPY values offset from the nominal position of the source, forming a streak along the direction of readout.

The [acisreadcorr](#) tool was designed to find these "out–of–time" events. Once flagged, the events may be

## Timing Analysis with Lightcurves – CIAO 3.4

filtered or corrected; see the [Remove the ACIS Readout Streak](#) thread for more information. Since the CHIPY value of an out-of-time event is related to its arrival time, a high-resolution lightcurve may be generated with the corrected photons.

The temporal resolution of this lightcurve is limited by the frame transfer rate of ~41 msec/frame. However, since the corrected time is based on the assumption that all photons originate from the center of the source, the error will be proportional to the size of the source. As mentioned in the [ahelp file](#), the accuracy of these times has not been rigorously tested, so use caution when performing timing analysis with these events.

### Precise, Flux-Calibrated Lightcurve

The most dependable method of calculating fluxes is to model the response-folded source spectrum. This will give an intrinsic flux [ergs/cm<sup>2</sup>/s], instead of a simple count rate [count/s]. Moreover, one may plot source spectral parameters (the slope of a power law, line FWHM, etc.) versus time for a more detailed study of the nature of the source variation. The [Create a Phase-binned Spectrum](#) thread shows how to extract the data.

In general, creating a flux-calibrated lightcurve is uncomplicated, but time-consuming. First, extract spectra for the times of interest; one spectrum is needed for every datapoint in the final lightcurve. Each spectrum is then fit with a [source model](#) in *Sherpa* and the integrated [flux](#) is calculated for a given energy range. Each flux value will be calibrated and based on the appropriate GTIs. Finally, the flux value for each of the fits can be manually written to a file for plotting in *ChIPS*; the file should have two columns — time and flux (the best-fit parameter value).

---

The Chandra X-Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.  
60 Garden Street, Cambridge, MA 02138 USA.  
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:  
<http://cxc.harvard.edu/ciao3.4/why/lightcurve.html>  
Last modified: 21 February 2007