

The EUVE Science Archive Data Products

The EUVE Permanent Archive

There are two components to the EUVE Permanent Archive: the EUVE Telemetry Archive and the EUVE Science Archive. The Telemetry Archive contains a complete copy of the raw telemetry from the entire EUVE mission, and is intended as a permanent record of the mission. It is not primarily designed to support scientific archival research and access to the data will be difficult because of the unprocessed nature of the telemetry. The EUVE Science Archive, however, is intended for scientific research and is what most people probably have in mind when they refer to the “EUVE archive”.

The EUVE Permanent Archive was created at the Center for EUV Astrophysics (CEA). The data products in the archive have been delivered from CEA to the NSSDC for distribution to the astronomical community. EUVE Science Archive pointed data products are being made available through archive sites at STScI and the HEASARC. At the time of this writing, EUVE continues to perform Guest Observer (GO) observations of new targets. The data from these observations is periodically delivered by CEA to the NSSDC and becomes available to the public through the archive sites.

Archive Documentation

This document details several aspects of the Science Archive and its use. It is intended as an introduction to the Archive for users who are already familiar with the EUVE GO data products that have been delivered since the start of the GO program. A more complete set of documentation will eventually become available when the *EUVE Guest Observer Software User's Guide* and *EUVE Guest Observer Data Products Guide* are rewritten to describe the Science Archive. Until those documents are revised, they are still useful for their in-depth discussions of data reduction and instrumental issues, but with regard to details about data formats or analysis steps, the descriptions in this introduction should be considered more up-to-date. Archive data users unfamiliar with the previous GO data products might want to consult the earlier documents to understand terms mentioned but not fully described here.

Note that the *EUVE Science Archive User's Guide* that is mentioned in the comments in the archive data headers refers to the rewritten *EUVE Guest Observer Data Products Guide* and therefore is not yet available.

The Science Archive Contents

The EUVE Science Archive contains multiple types of data. During the first six months of the EUVE mission in 1992–1993, a survey of the EUV sky was performed. The maps and catalogs from this survey have been either published in the astronomical literature and/or made available from CEA for several years. These products are part of the Science Archive but are not discussed further in this introduction. Since the end of the survey, EUVE has been performing pointed GO observations. These observations are primarily done with the Deep Survey/Spectrometer (DS/S) telescope; a very small number of pointed observations have been done with the Scanner telescopes. The Science Archive contains event lists and images from all these pointed observations.

Data from pointed observations is delivered at the time of the observation to the GO to whom it belongs. In the past, after the expiration of the proprietary period of the data, CEA also made exact copies of the data sets available to archival researchers. All of the data products delivered from CEA were in a standard format (a mixture of FITS tables, FITS images, and IRAF QPOE files), now called the GO format. Between March 1997 and February 1998, all existing EUVE pointed observations were reprocessed at CEA and data sets were produced in a new format, the Science Archive pointed format. The reprocessing produced a uniformly processed set of observations for the Science Archive.

A side effect of the reprocessing is that there is not a one-to-one correspondence between the observations in the Science Archive and those delivered in the past to GOs and archival users. Some observations were

split in pieces, while others were merged together. Where these differences exist, the new Science Archive definitions of the observations are to be considered more accurate (or at least more consistent).

The Science Archive Pointed Format

The Science Archive pointed format (also called the archive format) is substantially simpler than the format in which Guest Observer products were previously shipped (the GO format). The archived nighttime data for a single observation consists of two FITS files, named `events.fit` and `images.fit`.

`Events.fit` contains the event and monitor data for the observation, while `images.fit` contains images and filtering data. Each FITS file contains multiple FITS image and/or binary table extensions, which will be described in detail below.

When daytime data is available for an observation, it is packaged in separate files named `events_day.fit` and `images_day.fit`. Daytime data is only available for the SW spectrometer, and only for observations taken before 15 March 1996 (a few earlier observations do not have day data).

Moving targets require more analysis steps than non-moving targets and contain some additional auxiliary data. The `events.fit` file for a moving target will contain two additional extensions with this additional data.

Occasionally, there is an anomaly during the observation or processing of a target that causes the resulting data to have unusual characteristics. When EUVE personnel noticed such an anomaly during the observation or its processing, written notes were appended to the FITS files in an additional binary table extension. We cannot ensure that the notes are comprehensive or always useful to researchers, but at least some information will be available if needed.

Next, we describe the contents of the archive FITS files in the nominal case. We use the abbreviations DS, SW, MW, and LW for the Deep-Survey imager and the Short-, Medium-, and Long-Wavelength spectrometers, respectively, and ScA, ScB, ScC for the Scanner A, Scanner B, and Scanner C telescopes.

Nominal contents of `events.fit`

Night DS/S data

- Primary HDU, empty

The header of the primary header and data unit (HDU) contains keywords that are globally applicable to this observation. These keywords are also repeated in each extension. We don't use the INHERIT keyword mechanism supported by some FITS readers (i.e., the INHERIT keyword is always set to the value "F"). The primary data array is always empty.

- Extension 1, BINTABLE, extname = "valid_times"

Contains the time intervals during which telemetry was present in this observation. The sum of the lengths of these intervals is also in the global keyword VALIDTIM. This is NOT the exposure time, because it does not account for times when the detectors were off. However, it does tell you an upper limit for the exposure time. There is no concept of exposure time for event list data in the absence of any time filters, because the possible arrival times for events are unrestricted. Therefore, no exposure times appear in the headers in `events.fit`. To see exposure times, look in `images.fit`.

- Extension 2, BINTABLE, extname = "ds_events"

Contains the DS event list for this observation.

- Extension 3, BINTABLE, extname = "sw_events"

Contains the SW event list for this observation.

- Extension 4, BINTABLE, extname = "mw_events"

Contains the MW event list for this observation.

- Extension 5, BINTABLE, extname = “lw_events”
Contains the LW event list for this observation.
- Extension 6, BINTABLE, extname = “quadrant”
Contains quadrant-specific information for this observation. Only quadrants for which data is present in the event lists are included (except in the sums), meaning only one quadrant per detector in the archive. The information given for each quadrant is quadrant counts, telemetered counts, summed quadrant counts over all detector quadrants, deadtime correction, and combined deadtime & primbsch correction.
- Extension 7, BINTABLE, extname = “adcnts”
Contains the detector A-D counts for each detector that has an event list in this observation.
- Extension 8, BINTABLE, extname = “orientation”
Contains aspect and position information for the EUVE spacecraft during the observation.
- Extension 9, BINTABLE, extname = “corrected_aspect”
Contains the corrected aspect for moving targets that have been corrected by an ephemeris. The extension is omitted if not needed.
- Extension 10, BINTABLE, extname = “ephemeris”
Contains the ephemeris used to correct the aspect for moving targets. The extension is omitted if not needed.
- Extension 11, BINTABLE, extname = “anomaly”
Contains a written description of any anomalies or other unusual characteristics of this observation that may impact the analysis of the data. The text is encoded as rows of a binary table with a single 80 character wide column. The extension is omitted if not needed.

The extension numbers are for the nominal case. The final three extensions are optional, and if data for any detector is, for some reason, not present for an observation, the corresponding extensions are simply omitted from the FITS files. Therefore, extension numbers may vary. However, the order of the extensions is always the same when they are present. Always check the contents of your FITS files before using them.

Day DS/S data

In daytime DS/S data, the format is the same as for night data except that only the SW spectrometer data is present. This means that only one event list extension (sw_events) is included and the quadrant and adcnts extensions contain only columns relevant to the SW spectrometer.

Scanner data

In scanner data, the format is the same as for night DS/S data except that the appropriate scanner detectors replace the DS and spectrometer detectors. There is never daytime scanner data.

Nominal contents of images.fit

Night DS/S data

- Primary HDU, empty
The header of the primary HDU contains keywords that are globally applicable to this observation. These keywords are also repeated in each extension. We don't use the INHERIT keyword mechanism supported by some FITS readers. The primary data array is always empty.
- Extension 1, IMAGE, extname = “ds”
Contains an image of the DS event list, made by binning the events in sky coordinates after applying filters.

- Extension 2, IMAGE, extname = “sw_night”
Contains an image of the SW event list, made by binning the events in wavelength coordinates after applying filters.
- Extension 3, IMAGE, extname = “mw”
Contains an image of the MW event list, made by binning the events in wavelength coordinates after applying filters.
- Extension 4, IMAGE, extname = “lw”
Contains an image of the LW event list, made by binning the events in wavelength coordinates after applying filters.
- Extension 5, BINTABLE, extname = “ds_limits”
Contains the filter limits used to produce the DS image in the ds extension from the event list.
- Extension 6, BINTABLE, extname = “sw_night_limits”
Contains the filter limits used to produce the SW image in the sw_night extension from the event list.
- Extension 7, BINTABLE, extname = “mw_limits”
Contains the filter limits used to produce the MW image in the mw extension from the event list.
- Extension 8, BINTABLE, extname = “lw_limits”
Contains the filter limits used to produce the LW image in the lw extension from the event list.
- Extension 9, BINTABLE, extname = “anomaly”
Contains a written description of any anomalies or other unusual characteristics of this observation that may impact the analysis of the data. The text is encoded as the rows of a binary table with a single 80 character wide column. The extension is omitted if not needed.

Day DS/S data

In daytime data, the format is the same as the night data except that only the SW spectrometer data is present; this means that only one image (named sw_day) and limits table (named sw_day_limits) are included.

Scanner data

In scanner data, the format is the same as for night DS/S data except that the appropriate scanner detectors replace the DS and spectrometer detectors. There is never daytime scanner data.

Description of Science Archive header keywords

This is a brief description of the header keywords that appear in the archive FITS files.

Standard FITS keywords

The following are standard FITS keywords with a standard meaning and are not described here: SIMPLE, BITPIX, NAXIS, NAXISn, EXTEND, DATE, ORIGIN, CREATOR, and END.

These are standard FITS keywords used to describe an extension and its structure (some are BINTABLE specific): XTENSION, PCOUNT, GCOUNT, TFIELDS, EXTNAME, TTYPE_n, FORM_n, TUNIT_n, TDISP_n, TLMIN_n, TLMAX_n, TDMIN_n, TDMAX_n, TNULL_n, and INHERIT.

These are standard FITS keywords used in images for scaling and world coordinate system information, or identification: BSCALE, BZERO, CTYPE_n, CRVAL_n, CRPIX_n, CDELT_n, CUNIT_n, and FILENAME.

EUVE Science Archive keywords

- **TELESCOP**
The name of the entire observatory, i.e., “EUVE”.
- **INSTTYPE**
The type of EUVE telescope/instrument that produced this data: “DS/S” or “SCANNER”.
- **INSTRUME**
The particular EUVE telescope/instrument that produced this data: “DS/S”, “ScA”, “ScB”, or “ScC”. In scanner data, this keyword varies for each event list.
- **DETNAM**
The EUVE detector that produced this data: “ScA”, “ScB”, “ScC”, “SW”, “MW”, “LW”, or “DS”. This keyword varies for each event list, since each detector's events are in a separate extension.
- **PHOTOND**
This is a numerical detector number. It ranges from one to seven, corresponding to each of the detector values possible for the keyword DETNAM.
- **TIMEDEL**
For tables containing rows at regularly spaced time intervals, found in the quadrant, adcnts, orientation, and corrected_aspect extensions, this is the nominal interval length.
- **OBJECT**
The name of the target. Note that many EUVE targets have been observed multiple times and were often proposed under differing names. A single name has been selected for each target and used consistently for all observations.
- **RA_OBJ, DEC_OBJ**
The coordinates of the target, in decimal degrees, taken from the EUVE database. These are typically the coordinates the original Guest Observer specified, or coordinates taken from the SIMBAD database. For the DS or scanner detectors, the processed image is centered at these coordinates. For a moving target, these are the mean of the ephemeris coordinates during the observation.
- **RA_PNT, DEC_PNT**
The median coordinates, in decimal degrees, at which the instrument was actually pointed during the observation. This will be different from RA_OBJ and DEC_OBJ if the observation was performed off-axis or for moving targets. Most EUVE observations are performed at least slightly off-axis to avoid the DS deadspot.
- **RA_PROC, DEC_PROC**
The coordinates, in decimal degrees, that were used to process the spectrometer data. These are derived from a centroid of the image of the target on the sky as seen in the DS detector. Not all targets have a visible DS image – if the centroid could not be computed, RA_OBJ and DEC_OBJ are used in the spectrometer processing. In moving targets, this is always the same as RA_OBJ and DEC_OBJ.
- **OBSERVER**
The original PI that had the rights to this observation. In some observations, multiple PIs were given the rights to the data, but only one PI is listed here. If this was a calibration observation without a PI, the keyword will have the value “EUVE”. In some cases, calibration targets also had PIs but are listed as “EUVE”.
- **DATE-OBS, TIME-OBS**
The date and time (GMT) of the start of the observation. Note that the year assumes a prefix of 19.
- **DATE-END, TIME-END**
The date and time (GMT) of the end of the observation. Note that the year assumes a prefix of 19.

- **OBS_MODE**
The spacecraft pointing mode during this observation. For archive data, this will always be “POINTING”. Other modes would apply, for example, in survey, or while the spacecraft was slewing between targets.
- **DITHER**
The dithering mode during this observation: “DITHERED”, “SPIRAL”, or “NONE”. Dithering is a procedure of slightly changing the pointing position during the observation to reduce the effects of fixed-pattern noise. There are two kinds of dithering done by EUVE: pointed dithers (including nodding) and spiral dithering.
- **DETMODE**
The detector coordinate conversion mode during this observation: “WSZ” or “XY”. The EUVE telemetry can contain raw WSZ coordinates or converted XY coordinates. This mode can change during an observation and is determined separately for each event. The value of this keyword represents the mode of the majority of the events in the observation (in all event lists) and not necessarily the mode of all events. The mode of an individual event can be determined by inspecting the PH (pulse height) column in the event list. Events in XY mode will have an undefined PH. The mode also affects the precision of time stamps on the events; see the discussion of the TIERRABS keyword.
- **OFF-AXIS**
A Boolean flag indicating whether or not this observation was done primarily off-axis. This does not include small off-axis deviations due to dithering or DS deadspot avoidance.
- **MOVING**
A Boolean flag indicating whether this was a moving target. If so, aspect correction will have been performed, and the `events.fit` file should contain the optional `corrected_aspect` and `ephemeris` extensions. When the target is moving, the meanings of the coordinate keywords are changed as described earlier.
- **DAYNIGHT**
The telemetry type included in this observation: “DAY”, “NIGHT”, or “BOTH”.
- **VALIDTIM**
The sum of the lengths of the intervals in the `valid_times` extension, representing the total amount of telemetry which is present. This is NOT the exposure time, because it does not account for times when the detectors were off. However, it does tell you an upper limit for the exposure time. There is no concept of exposure time for event list data in the absence of any time filters, because the possible arrival times for events are unrestricted. Therefore, no exposure times appear in the headers in `events.fit`. In `images.fit`, the images have been made from filtered event lists, and the exposure times appear in the EXPTIME keyword.
- **RA_UNIT, DEC_UNIT**
The units of the RA and DEC values that appear in other keywords. Always has the value “deg” in the archive.
- **EQUINOX**
The equinox used for specifying coordinates in this observation. Always set to “2000.” in the archive.
- **RADECSYS**
The coordinate reference frame for this observation. Always set to “FK5” in the archive.
- **TIMESYS**
The time system used to specify times in this observation. In the archive, this refers to the TIME columns that appear in a number of the tables, as well as the TSTART and TSTOP keywords. The

value is always “MJD”. Note, however, that EUVE times are not simply the MJD, but rather the number of seconds since some specific MJD; see the MJDREF keyword below.

- **TIMEZERO**
An offset to be added to all EUVE event times to get their true time. This always has the value “0.” in the archive, meaning that EUVE times are already in the time system described by the keywords TIMESYS, TIMEUNIT, MJDREF.
- **TIMEUNIT**
The units of the EUVE times specified in this observation. This always has the value “s”, meaning all EUVE times are in seconds since the time specified in the keyword MJDREF.
- **CLOCKCOR**
A clock correction flag. Always has the value “NO” meaning that EUVE times are not guaranteed to be corrected to UT. Regular corrections are applied to the EUVE spacecraft clock to keep EUVE times near UT, but there is drift between corrections. The EUVE time is always kept within 0.9 msec of UT, and usually within 0.7 msec.
- **TIMEREF**
Another time correction flag. Always has the value “LOCAL”, indicating that EUVE times are in the time frame of the satellite, and are not corrected to the solar system barycenter or anywhere else.
- **TASSIGN**
The location where times are assigned to EUVE event detections. Always has the value “SATELLITE”, meaning that the times are assigned on the satellite.
- **TSTART, TSTOP**
The start and end time of the observation, corresponding to the DATE_OBS, TIME_OBS, DATE_END, and TIME_END keywords described earlier, but in the EUVE time system rather than UTC.
- **TIERRABS**
The precision of the EUVE times. For time tags on events, this depends on the coordinate conversion mode of the event (see the DETMODE keyword). WSZ mode events have time tags precise to 0.001 s, while XY mode events are only known to 0.008 s. In an event list, the TIERRABS keyword will have the value “0.001” or “0.008”, depending on the DETMODE of the observation, but remember that timing of an individual event depends on the mode of that event. For a non-event-list table, the precision is always 0.001 seconds. The absolute EUVE time frame can drift slightly relative to UT, but the maximum drift is always smaller than the precision of the time tags; see the notes on the CLOCKCOR keyword above.
- **MJDREF**
The reference time for all EUVE times. This always has the value “40000.” for the archive, meaning that the EUVE times are in seconds relative to a zero point at MJD = 40000 (JD = 2440000.5, or 24.00 May 1968).
- **EGOCSVER**
The EUVE processing software version used to produce this FITS file.
- **REFVERS**
The EUVE reference calibration data set used to produce this FITS file.
- **EXPTIME**
The exposure time for this image. This keyword appears in `images.fits` only and gives the exposure time for the image, corrected for the effects of primbsching and downtime.

- RAWEXP

The raw exposure time for this image. This keyword appears in `images.fit` only and gives the exposure time for the image, NOT corrected for the effects of primbsching and deadtime.

Description of FITS extensions

This is a description of the structure and meaning of all of the binary table and image extensions which can appear in Science Archive FITS files. No single FITS file will contain all these extensions. A table describing the columns of each binary table extension is given. The “Type” field refers to the FITS standard data types.

There are several categories of data in the binary table extensions. Event lists contain a row for every event detected by the EUVE detectors. Auxiliary values are reported in the telemetry at regular intervals during the observation, or derived from other auxiliary values or the event data. These values describe the state of the spacecraft or instruments. Other possibilities are time intervals, filter limits, or an ephemeris.

Events.fit extensions

`valid_times`

This is a binary table extension containing the time intervals for which telemetry was present during this observation. Each row contains the starting and ending times of a single time interval. The total time contained in all the intervals in this extension appears in the keyword `VALIDTIM`. The presence of telemetry does not imply that data is necessarily present in any detectors; for example, one or more detectors might be turned off. Therefore, these intervals do not represent the exposure time of the observation. The actual exposure time can only be defined by restricting the event lists to time intervals when the detectors were turned on. This is done during the processing of EUVE data when time filters based on A-D counts are applied in order to construct the images that appear in `images.fit`. See the discussion at the `adcnts` extension and the description of the `EXPTIME` keyword.

Column	Type	Units	Description
START	D	seconds	Starting time of interval.
STOP	D	seconds	Ending time of interval.

`ds_events`, `sca_events`, `scb_events`, `scc_events`

These are binary table extensions containing event lists for the EUVE imaging instruments. A separate extension is present for each detector that contained data for the observation. Each row of the table represents a single detected event.

Column	Type	Units	Description
TIME	D	seconds	Time of the event.
RA	D	degrees	Right ascension of the event.
DEC	D	degrees	Declination of the event.
DETX	I	pixel	Detector x-coordinate of the event.
DETY	I	pixel	Detector y-coordinate of the event.
PH	I		Pulse height of the event. This is only defined if the detector was in WSZ mode when this event arrived. See the discussion of the <code>DETMODE</code> keyword.

sw_events, mw_events, lw_events

These are binary table extensions containing event lists for the EUVE spectrometers. A separate extension is present for each detector that contained data for the observation. Each row of the table represents a single detected event.

Column	Type	Units	Description
TIME	D	seconds	Time of the event.
WAVELENGTH	E	Angstroms	The wavelength of the event.
ANGLE	E	degrees	The imaging angle of the event.
DETX	I	pixel	Detector x-coordinate of the event.
DETY	I	pixel	Detector y-coordinate of the event.
PH	I		Pulse height of the event. This is only defined if the detector was in WSZ mode when this event arrived. See the discussion of the DETMODE keyword.

quadrant

This binary table extension contains quadrant-specific auxiliary information for this observation, reported every 2.048 seconds, with occasional gaps in the data. The values in each row apply to the 2.048-second interval ending at the time of the row.

The columns that are present in this table vary; values are only included for quadrants which have potentially produced events in the event lists for this data set. A quadrant is (for our purposes) just a region of a detector over which quadrant counts are counted. EUVE detectors have one, two, three, or four quadrants. In most archive data sets, only one quadrant appears in this table for each detector: the quadrant that contains the source being observed. This is possible because in the DS/S instruments, sources are always observed in the same quadrants. These quadrants are SW: 0, MW: 1, LW: 1, and DS: 1. For pointed scanner observations, however, we cannot know the quadrant in which the source appears. Therefore, scanner data sets contain auxiliary values from all four quadrants for each detector present. The table below shows the columns you would expect to see in a DS/S observation with all detectors in use.

Quadrant counts are the total number of events detected in a detector quadrant during each 2.048-second interval, as determined on the spacecraft. The telemetered counts are the total number of events in a quadrant that appear in the telemetry as received on the ground. These two counts will differ due to loss of events during the telemetering process, typically due to insufficient bandwidth. The procedure by which events are selected for telemetering is called primbsching and the ratio of the quadrant counts to telemetered counts is called the primbsch correction. The summed quadrant counts are the sum of the individual quadrant counts for all quadrants of the detector, including those that do not appear in the table. This sum is used to determine the deadtime correction, a measure of the loss of events onboard the spacecraft due to deadtime in the processing hardware and software. Finally, the combined correction is the product of the deadtime and primbsch corrections.

Column	Type	Units	Description
TIME	D	seconds	Time for this row.
SWQ0SF	I	count	The quadrant counts in the SW quadrant 0.
SWQ0TM	I	count	The telemetered counts in the SW quadrant 0.
SWSUMQ	J	count	The summed quadrant counts over all SW quadrants.
SWDC	E		The SW deadtime correction.
SWQ0DPC	E		The combined SW deadtime and primbsch correction.
...	The last five rows repeat for every detector that is present in this observation, typically in the order SW, MW, LW, DS.

adcnts

This binary table extension contains auxiliary data that applies to an entire detector, reported every 1.024 seconds, with occasional gaps in the data. The values in each row apply to the 1.024-second interval ending at the time of the row.

The columns that are present in this table depend on which detectors are present in this data set. There will be one column containing A-D counts for each detector present

The A-D counts are hardware counts of the detections on each detector. They will differ from the summed quadrant counts for the same time interval because of filtering of events by the pulseheight threshold and loss of events due to deadtime in the detector electronics. A-D counts are used to detect whether a detector is on or off; three or more counts in a 1.024-second interval is normally taken to indicate that the detector is on. A-D counts can also be used for data-quality filtering, such as removal of times of high background.

Column	Type	Units	Description
TIME	D	seconds	Time for this row.
SWADCT	I	count	The A-D counts in the SW spectrometer.
MWADCT	I	count	The A-D counts in the MW spectrometer.
LWADCT	I	count	The A-D counts in the LW spectrometer.
DSADCT	I	count	The A-D counts in the DS detector.

orientation

This binary table extension contains auxiliary data describing the position and orientation of the EUVE spacecraft, reported every 1.024 seconds, with occasional gaps in the data. The values in a row apply at the instantaneous time of the row. Note that the row times in the orientation table are not exactly the same as those in the adcnts table.

The columns in this table are independent of which detectors have data in this data set, and are always as shown below.

The orientation information appears as the four components of a quaternion (the aspect) which represents the rotation from J2000 equatorial coordinates to a coordinate system fixed with respect to the spacecraft. For more details of the use of the aspect quaternion and additional references, see Abbott, *et al.* 1996, *ApJS*, **107**, 451. If this is a moving target and the aspect has been corrected for the motion of the source on the sky, an additional binary table extension named `corrected_aspect` will be present in this data set. The position information appears as the Cartesian coordinates of the spacecraft in the J2000 equatorial frame, in units of meters. The length of the position vector yields the spacecraft altitude.

Column	Type	Units	Description
TIME	D	seconds	Time for this row.
ASPECTW	D		Quaternion component.
ASPECTX	D		Quaternion component.
ASPECTY	D		Quaternion component.
ASPECTZ	D		Quaternion component.
POSITIONX	D	meters	J2000 equatorial spacecraft position component.
POSITIONY	D	meters	J2000 equatorial spacecraft position component.
POSITIONZ	D	meters	J2000 equatorial spacecraft position component.

corrected_aspect

This optional binary table extension is equivalent to the aspect portion of the orientation extension, except that the aspect values have been corrected for the motion of the source on the sky. This extension only appears for moving targets. The correction uses the source position information in the ephemeris extension.

Column	Type	Units	Description
TIME	D	seconds	Time for this row.
ASPECTW	D		Quaternion component.
ASPECTX	D		Quaternion component.
ASPECTY	D		Quaternion component.
ASPECTZ	D		Quaternion component.

ephemeris

This optional binary table extension contains an ephemeris for the source during this observation. This extension only appears for moving targets. The values on each row apply to the instantaneous time of that row. The frequency of tabulated points will vary.

The positions in this ephemeris represent the J2000 coordinates of the target as seen from EUVE, including any significant parallax effects. These positions are used to correct the aspect values for this observation to account for the motion of the source on the sky. The corrected values appear in the corrected_aspect extension.

Column	Type	Units	Description
TIME	D	seconds	Time for this row.
RA	D	degrees	Target right ascension.
DEC	D	degrees	Target declination.
DIST	D	AU	Target geocentric distance.

Images.fit extensions

ds, sca, scb, scc

These are image extensions containing images constructed by binning the detector event lists in remapped (sky) coordinates. RA is increasing along the x-axis and Dec is increasing along the y-axis. The event data is filtered using time filters constructed from the limits specified in the corresponding table extension (ds_limits, etc.).

sw_night, mw, lw

These are image extensions containing images constructed by binning the detector event lists in remapped (spectral) coordinates. Wavelength is increasing along the x-axis and imaging is increasing along the y-axis. The event data is filtered using time filters constructed from the limits specified in the corresponding table extension (sw_night_limits, etc.).

ds_limits, sw_night_limits, mw_limits, lw_limits

These binary table extensions contain limits on auxiliary data values that are used to filter event data for the observation when constructing images. This filtering removes earth-blocked times, times of high background, and times when the detectors were turned off.

Column	Type	Units	Description
NAME	A		Name of limited value.
LOW	E	unknown	Lower limit.
HIGH	E	unknown	Upper limit.

Common extensions

anomaly

This optional binary table extension contains a varying number of rows of text describing any anomalies in the data that were noticed during the observation or processing of this target. The extension only appears when needed. The comments are subjective in nature and are not guaranteed to be complete or useful.

Column	Type	Units	Description
TEXT	A		

Working With Science Archive Data

The archive format has been designed primarily as a FITS format. That means that considerable care has been taken to make the FITS files well organized and easily understood, and to provide complete meta-data in the headers. This means that the data is now much easier to read and manipulate using ordinary FITS tools. This is especially helpful to users who wish to bypass the IRAF file formats used in the EUV package and do their analysis in a non-IRAF environment. The description of the headers and tables given earlier should be sufficient for such users.

On the other hand, those users who wish to use the EUV package, or simply return to more familiar products, will have to reconstruct those products from the archive FITS files. Here are step-by-step instructions to unpack your data and reproduce the basic analysis steps, using IRAF and the EUV software package. The resulting products are very similar to the products that were delivered in the old GO format.

Note that the new procedure is rather different from the one you may have used in the past to unpack GO format data. In particular, you no longer need to use the EUV package task **qprst**.

IRAF releases and the EUV package

At the time of this writing, the most recent publicly available version of the EUV software package is version 1.7. That version of the EUV package is compatible with IRAF 2.10.4, but not with the most recent IRAF release, 2.11. Release 1.8 of the EUV package will work with IRAF 2.11.

This document is applicable to either version of the EUV package. In the instructions below, where parameter lists are shown for IRAF tasks, they will be the IRAF 2.11 versions (any differences are minor). The same instructions can be used with IRAF 2.10.4/EUV 1.7, but you need to refer to unpacked ST tables and IRAF images, as described in the next section, rather than FITS extensions.

Referring to tables and images

If you are using IRAF 2.11/EUV 1.8, you have several options available regarding the method you use to access the tables in your data set. Under IRAF 2.11, image and table processing tasks can directly read FITS image and binary table extensions. Several software tasks described below require tables from the data set as input. You can either unpack the FITS files into a one or more standalone ST tables, which you then specify as input where needed, or you can specify individual FITS binary table extensions directly as input to the software.

If you choose to use the FITS extensions directly, you have two ways to refer to a specific extension: by number, or by name. The extensions are numbered starting at one. This is probably most easily illustrated with a specific example. Let's assume you have unpacked your event file `events.fit` into individual ST tables using the method described below (the FITS file is assumed to have the nominal DS/S night contents described earlier). You therefore have an ST table named `ds_events.tab` in the current directory (as well as several other tables) and you also still have the original FITS file. You now need to specify the DS event list to a program. The following are all legal references to the DS event list:

<code>ds_events.tab</code>	Read the unpacked ST table.
<code>ds_events</code>	Same as the previous example. The “.tab” filename extension is assumed if not present.
<code>events.fit[extension=2]</code>	Read directly from the second extension in the FITS file.
<code>events.fit[2]</code>	Same as the previous example. A number is assumed to refer to an extension number.
<code>events.fit[extname=ds_events]</code>	Also reads from the second FITS extension, this time selecting the extension by name.
<code>events.fit[ds_events]</code>	Same as the previous example. Any word not otherwise meaningful is assumed to refer to an extension name.

We will always use the last of these methods of specifying a table in the code examples below. You can choose another if it is more convenient. Also, keep in mind that the files don't have to be in the current directory; you can specify the path to the file if needed.

There is one additional complication: the IRAF CL will try to interpret certain characters specially if you do not prevent it. In particular, it won't like the “=” (equals sign) character in the above examples. So when typing an expression on the command line, you either can quote the entire expression (using single or double quotes), or escape the equals sign, as in

```
'events.fit[extname=ds_events]'
```

or

```
events.fit[extname\=ds_events]
```

Of course, you can also avoid this problem by using one of the table references that doesn't contain an equals sign, or by entering your parameters in the IRAF **epar** utility, where they are not interpreted by the CL.

Important IRAF 2.10.4 note: If you are using IRAF 2.10.4, directly reading from a FITS file is not possible. You must always unpack your FITS files as described below and then refer to the ST tables using one of the first two methods in the above table.

Unpacking your FITS files

Convert the FITS extensions into ST tables and IRAF images. This step is optional when using IRAF 2.11/EUV 1.8, but required under IRAF 2.10.4. The conversion can be done using the **strfits** task in the TABLES package. Set up your parameters like this:

<code>fits_files = ""</code>	FITS data source
<code>file_list = ""</code>	File list
<code>iraf_files = ""</code>	IRAF filename
<code>(template = "none")</code>	template filename
<code>(long_header = no)</code>	Print FITS header cards?
<code>(short_header = yes)</code>	Print short header?
<code>(datatype = "default")</code>	IRAF data type
<code>(blank = 0.)</code>	Blank value
<code>(scale = yes)</code>	Scale the data?
<code>(xdimtogf = no)</code>	Transform xdim FITS to multigroup?
<code>(oldirafname = yes)</code>	Use old IRAF name in place of iraf_file?
<code>(force = yes)</code>	Force conversion from fits?
<code>(offset = 0)</code>	Tape file offset

The essential parameter is *oldirafname*, which should be set to “yes”, causing the newly created tables and images to have the same names as the FITS extensions they came from. Change to a directory where you want the unpacked products to appear. To unpack all of the extensions in a FITS file, run **strfits** using command lines like

```
cl> strfits events.fit 1 ""
```

and/or

```
cl> strfits images.fit 1 ""
```

where you replace the pathname of the FITS files with whatever is correct for the observation in your system. The FITS files need not be in the current directory; simply supply the full path (they can even be on a CDROM). Obviously, if you don't require both the event lists and images you need only unpack one of the FITS files.

To unpack only a single extension, specify the desired extension by number, such as

```
cl> strfits images.fit[1] 1 ""
```

Alternatively, you can simply copy the extension out using the **tcopy** task for a table, or the **imcopy** task for an image, which also conveniently allows you to specify the extension by name:

```
cl> tcopy events.fit[ds_events] ds.tab
```

```
cl> imcopy images.fit[ds] ds.imh
```

In addition to producing an ST table for each BINTABLE extension and an IRAF image for each IMAGE extension in the FITS files, **strfits** will leave behind an image named `tmp001.imh` or something very similar. This is an empty image representing the contents of the Primary HDU, which contains no data in EUVE archive files. You can safely delete that image if you wish.

Building IRAF data files

1. Reconstruct a QPOE file for an event list. At this point, you have either unpacked an event list from `events.fit` using the previous step, resulting in an ST table, or you are running IRAF 2.11/EUV 1.8 and you are working directly from the FITS file. The reconstruction is done using the task **cep** in the EUV package.

We assume that you have a subdirectory (or a link to a directory) in your current directory named *reference* that contains an installed copy of the EGODATA reference data set. You must have EGODATA version 1.15 or later, since that version contains new **cep** scripts for working with archive data.

Spectrometer data

To build a spectrometer QPOE file, set up the parameters for the task **cep** as follows:

database = "reference/archpipe"	Processing text database name
pipeline = "Rebuild SW Spectrometer Archive QPOE"	Pipeline record name
(ra = INDEF)	Source right ascension
(dec = INDEF)	Source declination
(raunits = "degrees")	Units for RA [degrees hours]
(tableindex = "")	Observation table index file
(events = "events.fit[sw_events]")	Observation event list table
(aspect = "")	Observation aspect table
(reference = "")	CEP reference file

The key parameters here are *database*, which we have set to point to a special pipeline database used for working with archive data; *pipeline*, which identifies the specific script **cep** will execute; and *events*, which identifies the input event list. All other parameters can be left at their defaults.

If you are building an MW or LW spectrometer QPOE, rather than SW, you need to change the *pipeline* parameter to refer to the appropriately named script, and change the *event* parameter to refer to the appropriate event list.

Non-moving imaging data

If you are building a non-moving imaging QPOE (DS or scanner), rather than a spectrometer, you need to change *pipeline* and *events*. You must also enter the coordinates at which the QPOE should be centered. The parameter setup for **cep** for a DS QPOE would resemble this:

database = "reference/archpipe"	Processing text database name
pipeline = "Rebuild DS Remapped Archive QPOE"	Pipeline record name
(ra = 123.45)	Source right ascension
(dec = 67.89)	Source declination
(raunits = "degrees")	Units for RA [degrees hours]
(tableindex = "")	Observation table index file
(events = "events.fit[ds_events]")	Observation event list table
(aspect = "")	Observation aspect table
(reference = "")	CEP reference file

You can enter any *ra* and *dec* at which you would like the QPOE centered. However, if you want to exactly reproduce the processing done at CEA, enter the values from the RA_OBJ and DEC_OBJ header keywords for the observation in question (dump the header of any table or image to see those).

For the scanner detectors, simply substitute the appropriate pipeline script: ScA, ScB, or ScC instead of DS.

Moving imaging data

If you are building an imaging QPOE (DS or scanner) for a moving target, the only change from a non-moving imaging target is that you should use the "SourceCentered" *pipeline* script, as shown in this example parameter setup:

database = "reference/archpipe"	Processing text database name
pipeline = "Rebuild DS SourceCentered Archive QPOE"	Pipeline record name
(ra = 0.)	Source right ascension
(dec = 0.)	Source declination
(raunits = "degrees")	Units for RA [degrees hours]
(tableindex = "")	Observation table index file
(events = "events.fit[ds_events]")	Observation event list table
(aspect = "")	Observation aspect table
(reference = "")	CEP reference file

In this case, you almost certainly want to supply an *ra* and *dec* of (0., 0.) so the center of the QPOE is the center of the source.

Those of you familiar with the EUV package and **cep** task will be glad to know that, when used for rebuilding a QPOE as described here, **cep** runs more quickly than during a reduction from raw data. This is because no calculations need to be performed. The running time will depend on the number of events in the event list.

The **cep** rebuild scripts do not support the rebuilding of multiple QPOE files at once because in this usage the input comes from a different file or extension for each event list; you will have to run **cep** once for each detector.

2. If you want to recreate the filters that were used to produce the images in the archive from the event lists, you will first need to build the earth blockage table using the EUV package task **backmon**. Assuming you have the EGODATA data set in the *reference* subdirectory as described in step 1, the parameters should be set up as

output = "backmon"	Name of output ST table
(orient = "events.fit[orientation]")	Name of input orientation ST table
(refdata = "reference/detector")	Name of ST table for reference data
(detectors = "ds")	Detector (for boresight quaternions)
(skip = 1)	Number of rows to skip over
(maxtimegap = 600)	Max gap in time (secs) for Sat. Vel. Vectors
(time = "time")	Time column of orientation table
(aspectw = "aspectw")	Aspect W input column name
(aspectx = "aspectx")	Aspect X input column name
(aspecty = "aspecty")	Aspect Y input column name
(aspectz = "aspectz")	Aspect Z input column name
(posx = "positionx")	Position X input column name
(posy = "positiony")	Position Y input column name
(posz = "positionz")	Position Z input column name

The default parameter values are set up for GO format products and must be changed slightly for archive format. The parameter *time* should be given the value "time". The parameter *detectors* can, for computing earth blockage, be set to any detector of the same instrument type, so "ds" is a valid value for any of the DS/S detectors. The parameter *skip* determines the time frequency of computed values—the value "1" will give the most accurate determination of earth blockage (one point per 1.024 seconds). The parameter *orient* should refer to the table containing EUVE position information.

Next, you use the EUV package task **dqselect** to generate the time filters. The parameters can be set up like this:

tables = "events.fit[adcnts],events.fit[quadrant],backmon"	Monitor Tables
(limits = "")	Limits Tables
(gap = 1)	Time Gap
(imode = "line")	Input mode
(format = "14.4f")	Interval format control
(device = "stdgraph")	Graphics device

where the important parameter is *gap*, which should have the value "1". This indicates that any intervals of missing data larger than the nominal interval spacing will cause a break in the filter.

Dqselect is an interactive task, so you can run it and use a session like this:

```
eu> dqselect
Command> limload images.fit[sw_night_limits]
Command> gtgen
Command> gtsave sw_gt
Command> q
```

This produces the table *sw_gt* containing the filter intervals when the detector was on and not earth-blocked, by filtering on the SW A-D counts and the DS/S look-zenith angle. If you want to see the particular monitor limits that were used to make the filter, dump the contents of the *sw_night_limits* table in *images.fit*. This example is for the SW spectrometer; you could do any other spectrometer by loading the appropriate limits file with the *limload* command and using an appropriate name with *gtsave*.

There appears to be a bug lurking in the **dqselect** program that causes it to sometimes fail when the *limload* command is run directly on a FITS file as in the example above. If this happens to you, the best solution is to unpack the limits file from the FITS file into an ST table before loading it into **dqselect**. For example,

```
eu> tcopy images.fit[sw_night_limits] sw_night_limits.tab
eu> dqselect
Command> limload sw_night_limits
```

3. Add the time filter to the QPOE file as a macro. This step makes the filter available for use in QPOE expressions. The task **tfilter** can be set up like this:

input = "sw_gt"	Input table name
(output = "sw.qp")	Output file
(name = "sw_gt")	Name of output macro
(start = "stime")	Name of starting column
(start = "etime")	Name of ending column
(cmdline = no)	Command line output flag
(format = "14.4f")	Interval format control

which will cause the ST table *sw_gt.tab* to be read and the intervals written in the QPOE file *sw.qp* as the macro named *sw_gt*.

If you want to see a list of the macros which are defined in your QPOE file, and their values, use

```
eu> qpmedit sw.qp * .
```

4. The final step is to create an image by binning the event list. The task **qpmkim** is used for that purpose.

<code>input = "sw.qp[time=(sw_gt)]"</code>	QPOE file
<code>output = "mysw.imh"</code>	Image file
<code>(corrtab = "events.fit[quadrant]")</code>	correction table name
<code>(refdata = "reference/detector")</code>	reference directory
<code>(applycorr = yes)</code>	apply correction
<code>(adjexptime = yes)</code>	adjust exposure time
<code>(errband = no)</code>	create error band
<code>(deadtime = yes)</code>	do deadtime correction
<code>(primbsch = yes)</code>	do primbsch correction
<code>(timecol = "")</code>	optional alternate time column name
<code>(corrcol = "")</code>	optional alternate correction column name
<code>(tepsilon = INDEF)</code>	alternate epsilon value used for time comparison
<code>(delttime = INDEF)</code>	alternate delta of correction times
<code>(detector = "")</code>	detector name
<code>(quadrant = INDEF)</code>	source detector quadrant [INDEF or 0-3]
<code>(infomsgs = yes)</code>	give processing information messages
<code>(fixmwcs = yes)</code>	apply QPOE mwcs workaround fix
<code>(timefield = "time")</code>	QPOE event time field
<code>(dxfield = "dx")</code>	QPOE event dx field
<code>(dyfield = "dy")</code>	QPOE event dy field

The *input* parameter specifies the QPOE file to bin and a time filter to apply. The parameter *corrtab* refers to the table containing the primbsch and deadtime correction values. **Qpmkim** will determine which detector it is processing from the QPOE header keywords, and use the appropriate column from the correction table. The exposure time will be written into the image header.

The resulting image will be exactly equivalent to the corresponding image in the *images.fit* archive file. We have noticed that a very small number of events are sometimes shifted by one pixel in the images rebuilt with the procedure described here, versus the image taken straight from the archive. The shift is always along the x-axis (RA) and is most likely caused by a roundoff error at some point in the processing. (By the time you run **qpmkim**, the event list has been converted from a QPOE to an ST table to a FITS binary table and back to a QPOE).

Changes From Previous Guest Observer Data

Comparison to GO format

Users familiar with the format previously used to deliver EUVE Guest Observer and Archive data (referred to as the "GO format") will initially find the new format to be rather different. The new format has been designed from the start as a FITS format for EUVE data. That means that all design decisions have been made with the goal of producing FITS files that are portable and compatible with other standards or conventions for delivering event data as FITS.

Within those restrictions, however, the format has been designed to be as backwardly compatible as possible with the GO format. This means that the FITS files can be unpacked to ST Tables and QPOE files can be rebuilt. The resulting set of files will resemble the equivalent files in the GO format. In addition, all tasks in the EUV/IRAF analysis package have been modified (as of version 1.7) to work with both GO & archive data formats with little change from the user's point of view.

More subtle than the data format changes are changes to the science content of the data. These have been made to try to accommodate the majority of scientific uses of the data while minimizing the size of the archive. The changes also attempt to maximize the utility of more reduced products such as images. This lessens the likelihood that a researcher will need to work with the less reduced event list products, since the latter are larger and more complex.

These are the most significant changes:

- The archive data products do not contain the complete raw telemetry for the observation. In particular, they do not contain all events that were detected during the observation. For those events which are delivered, we still provide both remapped and detector coordinates, so that the data may still be reprocessed. However, in the GO format, it was possible to go to the raw data in the ST Tables to get access to events that were stripped out of the QPOE files. This is no longer possible. For example, there is no access to the stimpins in the spectrometers.
- In the spectrometers, we still deliver only events in a strip in remapped coordinates. The strip has now been narrowed from Y=720:1299 to Y=875:1174 (about 22 arcmin). For extended sources for which the strip would render the data less useful, we make an exception and open the strip to Y=1:2048 (the whole field of view).
- In the DS data, while we still process only events in the Lexan quadrant, we no longer deliver only a horizontal strip of events in the range Y=720:1299 in remapped coordinates. All events in the Lexan quadrant are now included in the event list and image.
- The vast majority of engineering monitors—the columns in tables 1–8 in the GO format—have been dropped. Only those monitors which are believed to be in common use for data analysis will be in the archive: aspect, position, quadrant counts, and A-D counts (along with primbsch/deadtime information).
- The monitors have been reorganized. The quadrant extension contains quadrant counts previously in table2 and primbsch and deadtime info previously in the primbsch table. The adcnts extension contains A-D counts previously in table1. The orientation extension contains aspect and position information previously in table1.
- The event list FITS extensions are sorted by time, whereas the QPOE files in the GO format were sorted by position.
- In event lists containing sky coordinates, we have changed the RA values to be in the range 0 to 360 degrees. In previous data, it was in the range -180 to 180 degrees.
- All event lists now contain pulse height information for each event for which it was available (meaning all events taken in WSZ mode).
- The header keywords in the tables and images have been completely reworked to meet FITS standards or guidelines from the HEASARC for event list data. In particular, the cards JDREF, OBSERVAT, and PI_NAME have been deleted. The cards INSTRUME, TELESCOP, TIMESYS, and ORIGIN now have different values. And many, many new cards have been added.
- The data types of some table columns have changed from floating point to integers: primarily all the counts in the quadrant extension.
- The DS image we produce has now been filtered with an upper limit on the quadrant counts in quadrant 1 (the Lexan filter). The upper limit is 50 counts per two frames for all but a handful of the brightest DS sources; for those, we increase it to 150. In GO format data, there was no upper limit in the DS count rate filter.
- For moving targets, we now use an ephemeris to correct the aspect for the motion of the target on the sky. The ephemeris and the corrected aspect values are included with the observation data set.
- For some moving targets, in addition to the strips of events mentioned above, we also remove events in certain portions of the detector specified in detector coordinates, such as the stimpins and unused quadrants. This is to prevent these detector features from being smeared all over the image of the source due to the large source motion on the detector.

Frequently Asked Questions

Why is the extension containing SW spectrometer data named `sw_night` in `images.fit`, but only `sw` in `events.fit`?

When the archive format was originally designed, we considered the possibility that the sites delivering archive data would wish to combine the night and day image data into a single FITS file. This could be convenient since those products are presumably the most commonly desired for any observation. In order to support this, we had to give the SW spectrometer day and night data different extension names. To date, no such combined delivery is being done, as far as we know, but it is still supported.

Where is the DISPAXIS header keyword that **euextract** complains is missing from my image?

The old GO format products had a header keyword named DISPAXIS in each spectrometer image. This keyword is used by some spectral analysis programs in IRAF to identify the orientation of the spectrum; for example, it is used in the extraction program **euextract** in the EUV package. Because the keyword was specific to certain analysis programs, we omitted it from the archive image headers.

If you need to use such a program, you can add the keyword to the image using the IRAF task **hedit**. For nominal EUV spectra, the value of the keyword should be "1", indicating that the spectrum lies along the x-axis. The **euextract** in particular has been modified in version 1.8 of the EUV package to have an additional parameter which allows you to specify the dispersion axis without having to add a keyword.

How do I see the ephemeris used in my moving target analysis before parallax corrections were applied?

Unfortunately, this is not possible. The archive products for moving targets include the ephemeris after the application of parallax corrections for the position of EUVE (applied using the EUV package task **parallax**). There is no simple way to remove those corrections. In hindsight, this was a poor decision. It would have been better to include the raw ephemeris, since users could always have reapplied the parallax corrections using the **parallax** task and the orientation table.

How can I rebuild QPOE files for all of the event lists at once?

You cannot. The rebuild scripts for **cep** are only capable of functioning on one event list at a time. This is a limitation of the **cep** program (it can only read from one table) and cannot be reasonably worked around.

Where are the extracted spectra?

No extracted spectra are part of the EUVE Science Archive. This was done because automatic extraction of spectra during the pipeline processing of observations at CEA is very unreliable. EUVE spectral images typically have a very low background, and many spectral sources are faint and are composed of discontinuous spectral lines. These features make it very difficult for tracing algorithms to find the spectra on the images. In addition, flexure of the instrumentation means we cannot know in advance precisely where the spectrum will fall on the image in every case. Because of these reasons, we do not automatically produce extracted spectra, and resource limitations did not allow hand extraction of spectra for the archive.

The archive sites may in the future make available quick-look spectra that are produced during processing using the assumption of no flexure and a perfectly straight spectrum. However, in many cases these spectra are known to be inaccurate. They also are not subject to the same quality control as the event and image data. If you obtain such spectra, please use them extreme caution—they are for quick-look purposes only and should not be used for analysis. You should always do science with spectra that have been hand-extracted from the images.

Mark Abbott
Center for EUV Astrophysics
26 June, 1998