SWIFT

TELDEF FILES : CONVENTIONS FOR ATTITUDE, ALIGNMENT AND COORDINATE TRANSFORMATIONS FOR USE IN SWIFT ANALYSIS SOFTWARE

VERSION 0.1

April 22, 2004

Prepared by Padi Boyd, Swift Science Center and Dean Hinshaw, Swift Data Center

TELDEF FILES: CONVENTIONS FOR ATTITUDE, ALIGNMENT AND COORDINATE TRANSFORMATIONS FOR USE IN SWIFT ANALYSIS SOFTWARE

1. Introduction

The purpose of this document is to define the coordinate systems used in Swift data analysis, the conventions necessary in order to specify the transformation between these coordinate systems, and a brief overview of how this is implemented in the software.

2. Coordinate Systems

2.1. Main Coordinate Systems

There are 3 types of coordinate systems which are of primary interest for use the Swift analysis software:

- The Spacecraft hardware coordinate system. This system is defined in the Swift Interface Requirements document, and has the spacecraft pointing direction (boresight) as the positive X axis. Unless otherwise specified, in this document this is the coordinate system being referred to when using the term "Spacecraft coordinates".

- The software Detector coordinate system for each of the instruments. These are the coordinate systems used for each of the detectors, and for Swift are centered on the detector in question, and have their positive Z axis as the pointing direction. Note that these coordinate systems are not the same as the hardware coordinate systems defined for each instrument. In this document we will simply refer to these as "Detector coordinates".

- The Sky coordinate system. This a planar projection of the celestial sphere, with axes fixed on the sky.

2.2. Ancillary Coordinate Systems

In addition, there are several other coordinate systems which are useful when analyzing data:

- Raw coordinates. These are coordinates specific to a particular detector and are generally defined in a way specific to the detector architecture. Some detectors also have additional sets of internal Detector coordinate systems, which for the purposes of this document are treated the same as Raw coordinates.

- Spacecraft Boresight coordinates. This coordinate system is co-aligned with the Spacecraft coordinate system, with the spacecraft boresight pointing along the positive Z axis. This is a reference coordinate system which is used for specifying how the Spacecraft coordinate system is oriented with respect to the boresight, which can vary from mission to mission. They are referred to in this document as the "Boresight coordinates".

- Mean NFI coordinates. This is a coordinate system which has its positive Z axis as the mean boresight of the narrow field instruments, ie the XRT and UVOT. They are referred to in this document as the "NFI coordinates".

3. Conventions

3.1. Reference coordinate system

In the Swift mission all coordinate transformations are referenced with respect to the Spacecraft coordinate system. The exception to this is the Raw coordinate systems, whose transformation is referenced to the Detector coordinate system of the respective instrument. Thus the Spacecraft attitude (which is stored as a quaternion), describes the transformation from Sky to Spacecraft coordinates. And the transformation to Detector coordinates (which is stored in a teldef file) is likewise a transformation from Spacecraft to Detector coordinates. See below for details on the use of teldef files. Similarly, Boresight and NFI coordinates are referenced to the Spacecraft coordinates using teldef files.

3.2. Ra, Dec and Roll

The attitude of the Swift spacecraft is reported in the telemetry as a triplet of numbers, whose values are the right ascension, declination and roll angle of the attitude of the spacecraft, ie with respect to the spacecraft boresight. In order to convert these values to quaternions, it is necessary to know the conventions used when specifying these values. In particular, there is an ambiguity in the roll angle, in that it can refer to a positive or negative rotation. For the Swift mission, the convention used is recorded as a keyword in the spacecraft alignment teldef file, and has a negative sign. There is also the possibility of having a roll angle offset, which is zero for Swift. For a more precise definition of Ra, Dec and Roll, see the Swift Spacecraft to Payload Telecommand ICD, (1143-EI-S22904).

4. Implementation

4.1. Teldef files

Teldef files are used to specify all coordinate transformations used in Swift analysis software. Teldef files are FITS files, and their format specification is included in Appendix 1. A spacecraft-level Teldef file is used to orient the spacecraft coordinate system to the internal software coordinate system. Instrument-level Teldef files describe the transformation from detector coordinates to spacecraft boresight coordinates. Examples of the spacecraft-level teldef file and instrument level Teldef files are given in Appendix 2. Some additional information (such as the roll angle convention) is also stored in these files. Note that not all the features of the teldef file convention are needed for all applications. For example, when specifying the transformation from Spacecraft to Boresight coordinates, only the ALIGNM keywords need be specified.

4.2. Examples of tools that use Teldef files

This section shows examples of software programs used by Swift to manipulate attitude information. A brief description is given of each tool and how it makes use of teldef files. The attitude library will stabilize a few months before launch, and a complete list will be included at that time.

4.2.1. acs2fits

This tool reads attitude information from the Swift telemetry, and creates an attitude history file, with quaternions and sky coordinates for each available attitude measurement. Because it reads data from the Swift telemetry, it is a Swift specific tool. The spacecraft telemetry contains ra, dec and roll values, which effectively specify the spacecraft attitude in the Boresight coordinate system. Thus in order to convert these values into quaternions referenced to the Spacecraft coordinate system, a teldef file is need to specify the transformation from the Boresight coordinate system to the Spacecraft coordinate system.

4.2.2. prefilter

This program is a generic tool which creates a table of orbit and attitude quantities at regular intervals, which can in turn be used for creating good time intervals for filtering event data. It takes as one of its inputs the attitude file output by asc2fits, and can produce filter columns of ra, dec and roll. Thus it needs a teldef file to transform quaternions into the coordinate system whose attitude is to be specified. Note the attitude does not necessarily have to refer to the Boresight coordinate system, but could also refer, for example, to the NFI Coordinate system.

4.2.3. attdump

This program takes as input an attitude file as above, and prints an ascii table of attitude values in ra, dec and roll. Similarly to prefilter, it needs a teldef file to convert quaternions into the coordinate system whose attitude is to be reported.

4.2.4. aspect

This program takes an attitude file and calculates a 'mean' attitude in sky coordinates, referenced to the coordinate system specified by the input teldef file. It is a generic tool.

4.2.5. coordinator

The coordinator program is a generic tool that takes an event list with locations specified in Raw coordinates, and calculates these positions in Detector and/or Sky coordinates. The input teldef file specifies the transformation from Raw to Detector coordinates, and from Detector to Spacecraft coordinates. The transformation to Sky coordinates is then completed using an attitude file as input.

4.2.6. imagexform

The imagexform program operates similarly to the coordinator program, except that it takes an image in Raw detector coordinates, and converts that image into Detector and/or Sky coordinates.

Appendix 1

1. Introduction

This is a copy of the *Teldef File Format Specification*, dated 2003-10-01, written by Ed Pier.

1.1 History

Telescope definition (teldef) files were first introduced to define instrument coordinate systems for the ASCA mission. The teldef format has since been generalized for other missions and is currently being used by Astro-E and Swift. This document specifies this format.

1.2 Concepts

All coordinate systems belong to one of three classes: raw, detector, or sky as described below.

1.2.1 Raw Coordinates

Raw coordinates give event locations in their most basic form - usually as they appear in the telemetry. A detector may be composed of a number of *segments* arranged arbitrarily in the focal plane. Each segment has its own raw coordinate system. For example, the ASCA SIS has four segments - one for each CCD chip. The Astro-E XRS has 32 segments - one for each XRS pixel. The two Swift XIS readout amps are represented by two overlapping segments. The mapping from raw coordinates to the focal plane may be linear or may have non-linear distortion corrections (e. g. the ASCA GIS and Swift UVOT).

1.2.2 Detector Coordinates

Detector coordinates give the location of an event within the focal plane. There may be any number of detector coordinates related to one another by linear transformations. Most instruments only have a single type of detector coordinate, however the Astro-E instruments have three: ACT, DET, and FOC.

The teldef format requires detector coordinates to be arranged in a particular order, such that raw coordinates are first transformed into the bottom level detector coordinates, and these are in turn transformed to the next level detector coordinates, etc.. One particular type of detector coordinate must be specified as being used in detector to sky transformations.

1.2.3 Sky Coordinates

Sky coordinates refer to positions in a tangent plane projection of the celestial sphere. The sky X coordinate is aligned with Right Ascension and the Y coordinate is aligned with Declination. Note that the attitude of the spacecraft is needed to transform from detector to sky coordinates.

2. File Format

A teldef file is a FITS file with most of its information given by keywords in a primary HDU with no data (i.e. NAXIS = 0). Additional extensions are sometimes used to specify the transformation from raw to detector coordinates.

Below we describe the FITS format in detail. Section 3. gives some examples of teldef files. It may be helpful to refer to those while reading this section.

2.1 General Keywords

The following keywords are required:

- TELESCOP - The name of the mission or spacecraft
- INSTRUME - The name of the instrument which this file describes

Other keywords are allowed as long as they do not conflict with the keywords described in this document. In particular, the CALDB requires a set of keywords.

2.2 Coordinate Names

Each coordinate type must have a unique three letter name in all caps. The raw coordinates must be named "RAW", and the sky coordinates must be named "SKY". The coordinate names are used in FITS keywords within the teldef file. Below, we will denote the coordinate name by "*xxx*" and use "[XY]" to mean "either X or Y".

The following keywords give the coordinate names:

- NCOORDS - This gives the number of coordinate systems, including raw and sky coordinates. (i.e. the number of detector coordinates + 2).
- COORDn - This gives the name of the "nth" coordinate system, with n=0 for raw coordinates, and n=NCOORDS-1 for sky coordinates.

For backwards compatibility with ASCA, if the above keywords are omitted, the default is:

```
NCOORDS =  3
COORD0  = 'RAW'
COORD1  = 'DET'
COORD2  = 'SKY'
```

Event files will have columns for each type of coordinate in the teldef file. These column names are specified as follows:

- *xxx*_[XY]COL - Optional keyword giving the event file column name.

If this keyword is omitted, it defaults to
*xxx*_XCOL = '*xxx*X'
*xxx*_YCOL = '*xxx*Y' .

except for sky coordinates, which default to
SKY_XCOL = 'X'
SKY_YCOL = 'Y'

2.3 Detector Coordinates

The following describes how detector coordinate systems and the transformations between them are specified.

2.3.2 Coordinate System Properties

The following keywords specify the properties of a single detector coordinate system.

- *xxx*_[XY]SIZ - The number of pixels in the coordinate address space. This keyword is required.
- *xxx*[XY]PIX1 - The coordinate of the center of the first pixel. This is usually either 0 or 1. This keyword is required.
- *xxx*_[XY]SCL - The physical size of each pixel. This keyword is required.
- *xxx*_UNIT - The physical units of *xxx*_[XY]SCL. This keyword defaults to "mm" if omitted.

2.3.3 Coordinate System Transformations

The following keywords specify the transformation between adjacent detector coordinate systems, with *xxx* being the name of the higher coordinate. The order of the coordinates is specified by the COORDn keywords as described in section 2.2. When *xxx* is the bottom level coordinate name, the keywords refer to the raw to detector transformation as described in section 2.4.3.

- *xxx*_[XY]OFF - These keywords give the offset of the center of the *xxx* coordinates with respect to the center of the next lower coordinate system. If omitted, the default is 0.
- *xxx*_SCAL - This gives a scaling between the next lower coordinate system and *xxx*. The scaling is applied such that the center of the *xxx* coordinates remains fixed.
- *Xxx*[XY]FLIP - These indicate whether or not the given axis should be inverted. a value of -1 indicates an inversion while a value of 1 indicates no inversion.
- *xxx*_ROTD - This keyword gives the rotation in degrees between *xxx* and the next lower coordinate system. The rotation is counter-clockwise about the center of the *xxx* coordinates

The above transformations are applied in the order: translation, rotation, scaling, inversion. So if the linear transformation from (X_0, Y_0) to (X_1, Y_1) is given by
$X_1 = A \cdot X_0 + B \cdot Y_0 + E$ and
$Y1 = C \cdot X_0 + D \cdot Y_0 + F$,
then
$A = \text{SCAL} \cdot \text{XFLIP} \cdot \cos(\text{ROTD})$,
$B = \text{SCAL} \cdot \text{XFLIP} \cdot \sin(\text{ROTD})$,
$C = -\text{SCAL} \cdot \text{YFLIP} \cdot \sin(\text{ROTD})$,

$D = \mathrm{SCAL} \cdot \mathrm{YFLIP} \cdot \cos(\mathrm{ROTD})$,
$E = C_{X1} + A\cdot(\mathrm{XOFF} - C_{X0}) + B\cdot(\mathrm{YOFF} - C_{Y0})$, and
$E = C_{Y1} + C\cdot(\mathrm{XOFF} - C_{X0}) + D\cdot(\mathrm{YOFF} - C_{Y0})$,
where the center of the (X_0, Y_0) coordinates is at (C_{X0},C_{Y0}), and the center of the (X_1, Y_1) coordinates is at (C_{X1},C_{Y1}).

2.4 Raw Coordinates

The following describes how the raw coordinate systems and the transformations to detector coordinates are specified.

2.4.1 Multiple Segments

There can be an arbitrary number of raw coordinate systems (segments). However, unlike the detector coordinates, which are arranged "above" one another, the raw coordinates are arranged "beside" each other. In other words, detector coordinates can be converted from one type to another, but raw coordinates can only be converted to the bottom level detector coordinates (i.e. the coordinates named by the COORD1 keyword).

The segments must be numbered sequentially, but unless otherwise noted, the number of the first segment is arbitrary. If there are multiple segments, an event file should have a column which specifies the segment number for each event. The name of this column is specified with the following keyword:

- SEG_COL - This keyword is required if there is more than one segment. However, for backwards compatibility, the default for ASCA SIS is "CCDID".

2.4.2 Coordinate system properties

The properties of the raw coordinate systems are specified using the keywords listed in section 2.3.2 with *xxx* set to "RAW". However, there are a few differences. Only one set of "RAW" keywords is possible, so all the raw coordinate systems must have the same dimensions, scaling and column names. The RAW_[XY]SCL keyword is optional with the default being no scaling (i.e. a scale of "1" and units "pixels"). However if both RAW_XSCL and RAW_YSCL are specified, the default units are "mm" just as for detector coordinates.

For backwards compatibility, ASCA teldef files have the following default values:

keyword	**SIS**	**GIS**
RAW_XSIZ	419	256
RAWXPIX1	6	0
RAW_YSIZ	416	256
RAWYPIX1	2	0

2.4.3 Raw to Detector Transformations

Different instruments have different natural ways of specifying the transformation from raw to detector coordinates. Therefore teldef files support different formats for these transformations.

The raw to detector transformation is a combination of two successive transformations. The first is a transformation from the raw coordinates to an arbitrary set of internal coordinates. The second is a transformation from the internal coordinates to the bottom level detector coordinates. Typically, the first transformation is used to describe the arrangement of detector segments, and the second is used to transform all the segments at once.

2.4.3.1 Internal to Detector Transformation

The transformation from internal to detector coordinates is given by the keywords listed in section 2.3.3, with *xxx* equal to the name of the bottom level detector coordinates. However, for this transformation, the inversion keywords default to *xxx*XFLIP=1 and *xxx*YFLIP=-1. This is because for most instruments the raw coordinates are "look down", while the detector coordinates are "look up".

Note that the internal coordinates do not have a set of keywords like those in section 2.3.2. Therefore the following keywords are necessary to define the center of the internal coordinate system.

- INT_[XY]CEN - Center of the internal coordinate system. If these keywords are missing, they default to the center of the bottom level detector coordinates.

2.4.3.2 Raw to Internal Detector Transformation

Below we list the currently available methods for describing the transformation from raw to internal coordinates. Only one of these methods should be used in a particular teldef file.

2.4.3.2.1 SEGMENT CORNER LIST

This method lists the locations of the four corners of each segment in a FITS BINTABLE extension. This method was used for the Astro-E XRS, since there is a large number of segments, and the pixel corner locations had been measured in the lab.

The extension must have EXTNAME= 'PIXEL_MAP'. There may be no other extensions with this name, even if this method is not being used.

The table must have the following columns:

```
TTYPE1  = 'PIXEL   '      / pixel ID number
TTYPE2  = 'X       '     / coordinate of pixel corners
TTYPE3  = 'Y       '     / coordinate of pixel corners
```

The PIXEL column lists the segment number. Note that for this method the first segment must be numbered "0", and there must be one and only one row for each segment. The X and Y columns must each contain 4 dimensional vectors listing the internal coordinates of the corners. The corners correspond to raw coordinates as follows:

Corner	RAWX	RAWY
0	RAWXPIX1 - 0.5	RAWYPIX1 - 0.5
1	RAWXPIX1 - 0.5 + RAW_XSIZ	RAWYPIX1 - 0.5
2	RAWXPIX1 - 0.5 + RAW_XSIZ	RAWYPIX1 - 0.5 + RAW__YSIZ
3	RAWXPIX1 - 0.5	RAWYPIX1 - 0.5 + RAW__YSIZ

In other words, they start at the lower left hand corner and go counter-clockwise around the segment with respect to the *raw* coordinate system.

2.4.3.2.2 LINEAR COEFFICIENTS

Another method of specifying the transformation from raw to internal coordinates is to explicitly list the coefficients of a linear transformation using keywords of the form COE_[XY]*n*_[ABC], where *n* is the segment number and may be omitted if there is only one segment. There must be one and only one set of coefficients for each segment. The number of the first segment is arbitrary. The transformation is as follows:

$$X_{int} = \mathrm{COE\_X}n\mathrm{\_B} \cdot X_{raw} + \mathrm{COE\_X}n\mathrm{\_C} \cdot Y_{raw} + \mathrm{COE\_X}n\mathrm{\_A},$$
$$Y_{int} = \mathrm{COE\_Y}n\mathrm{\_B} \cdot X_{raw} + \mathrm{COE\_Y}n\mathrm{\_C} \cdot Y_{raw} + \mathrm{COE\_Y}n\mathrm{\_A},$$

where (X_{int}, Y_{int}) are the internal coordinates and (X_{raw}, Y_{raw}) are the RAW coordinates.

2.4.3.2.3 ASCA GIS

The ASCA GIS teldef files have a few quirks which require special handling. Therefore, this format is not documented here.

2.4.2 Non-linear Distortion Corrections

The raw to detector coordinate transform may contain a non-linear component represented by a map of correction values. The distortion map is specified in a separate FITS extension in one of two formats described below. If the instrument has only one segment, the map may be given in either raw or detector coordinates. If the instrument has multiple segments, the map must be in the bottom level detector coordinates. The following boolean values keyword indicates the map coordinates:

- CORINRAW - If "T" the map is in raw coordinates, if "F", it is in the bottom level detector coordinates.

2.4.2.1 Image method

For this method the map is given as a FITS image data cube with NAXIS=3. The data are stored in an image with two planes (i.e. NAXIS3=2). The first plane gives the corrections in X and the second plane gives the corrections in Y. The standard WCS keywords

(CRPIXn, CRVALn, and CDELTn) relate the first two axes to the raw or detector coordinates. Note that the image can sample the raw or detector coordinate space arbitrarily.

2.4.2.2 Table method

For this method the map is represented by a table listing corresponding positions before and after the corrections are applied. The locations of these points are arbitrary. The table must have the following columns:

```
TTYPE1  = 'X      '        / uncorrected X position
TTYPE2  = 'Y      '        / uncorrected Y position
TTYPE3  = 'NEWX   '        / corresponding corrected X position
TTYPE4  = 'NEWY   '        / corresponding corrected Y position
```

These points are interpolated to a regular grid by a method specified in the following keyword:

- METHOD - Currently the only allowed value is "distance" which indicates distance-weighted interpolation.

The regularly spaced grid may subsample the raw or detector coordinates by a factor given by the following keywords:

- SAMPLE[XY] - gives the number of raw or detector pixels per grid pixel in the X and Y directions.

2.5 Sky Coordinates

Sky coordinates give the position of an event on the sky as projected onto a tangent plane. The sky X axis points in the negative right ascension direction, and the sky Y axis points in the positive declination direction.

2.5.1 Originating Detector Coordinates

The sky coordinates must be associated with a particular set of detector coordinates. The pixel size and shape is assumed to be the same in the sky coordinates and the originating detector coordinates. If there is more than one type of detector coordinate, then the originating coordinates must be specified with the following keyword:

- SKY_FROM - This gives the name of the associated detector coordinates.

2.5.2 Sky Coordinate Properties

The keywords listed in section 2.3.2 also apply to sky coordinates with *xxx* equal to "SKY". However there are a few differences from their use for detector coordinates. The SKY_[XY]SIZ and SKY[XY]PIX1 keywords are optional. If they are omitted, they default to the corresponding values of the originating detector coordinates. The SKY_[XY]SCL keyword is not allowed, since the sky pixel scale is set by the size of the originating detector coordinates and the focal length. The SKY_UNIT keyword may have one of the following values: "arcmin", "deg", "rad", indicating arc minutes, degrees, or radians, respectively. The default value is "arcmin".

2.5.3 Detector to Sky Transformation

2.5.3.1 Plate Scale

The focal length of the telescope must be specified by the following keyword:

- FOCALLEN - Focal length of the telescope in the same units as the pixel scale (*xxx*_UNIT) of the originating detector coordinates.

The number of pixels per radian is equal to FOCALLEN/*xxx*_[XY]SCL, where *xxx* is the coordinate name in the SKY_FROM keyword.

2.5.3.2 Alignment

The conversion from detector to sky coordinates is composed of two parts: the alignment of the detector coordinates with the spacecraft axes and the orientation of the spacecraft in space. The second part is generally a function of time, but the first part must be specified in the teldef file using the following keywords:

- ALIGNM*nm* - Components of 3x3 alignment matrix giving the orientation of the the spacecraft axes with respect to the detector coordinates.

where *n* and *m* have values 1, 2, or 3 corresponding to X, Y or Z axes.

The detector to sky transformation is performed by assuming that the detector coordinates are perpendicular to the telescope axis and that this axis is situation directly over the center of the detector coordinate system. We then have a three dimensional coordinate system formed by the detector X and Y axes and with a Z axis pointing along the telescope. Note that the detector coordinates must be "look up". In other words these three dimensional detector axes must form a right-handed coordinate system. Then the rotation from these 3-D detector coordinates to the spacecraft axes is given by ALIGNM*nm* as follows:

SPCX = ALIGNM11 · DETX + ALIGNM12 · DETY + ALIGNM13 · DETZ
SPCY = ALIGNM21 · DETX + ALIGNM22 · DETY + ALIGNM23 · DETZ
SPCZ = ALIGNM31 · DETX + ALIGNM32 · DETY + ALIGNM33 · DETZ ,
where (DETX, DETY, DETZ) are the three dimensional detector coordinates and (SPCX, SPCY, SPCZ) are the spacecraft axes.

Note that in reality the above assumptions are not exactly correct. This means that any mis-alignment between the telescope axis and the detector coordinate center is approximated as a tilt of the telescope axis with respect to the spacecraft axes. This approximation is accurate to the square of the size of this fictitious tilt in radians. Even an extreme 1 degree mis-alignment would still be accurate to one part in ~10,000.

2.5.3.3 Roll Angle

Occasionally, it is useful to specify an orientation using Right Ascension, Declination, and Roll Angle. The convention to use for roll angle is specified by the following optional keywords:

- ROLLSIGN - Gives the sign of the roll angle, either -1 or +1. Defaults to +1.
- ROLLOFF - Gives the offset of the roll angle in degrees. Defaults to 0.

The default roll angle convention is such that the detector X axis is aligned with the sky X axis when roll=0. A positive roll angle rotates the detector coordinates clockwise with respect to the sky coordinates.

if an orientation is specified as a set of Z-Y-Z Euler angles, with a third Euler angle, _, in degrees, then the roll angle is given by

roll = (_ - 90 + ROLLOFF) * ROLLSIGN

2.5.4 Optical Axis Location

As described in section 2.5.3 the position of the optical axis is not explicitly used in the transformation from detector to sky coordinates. However, it is important for other purposes, like calculating the telescope response at a given location in detector coordinates. The following keywords give the position of the optical axis in the detector plane:

- OPTCOORD - The name of the coordinate system used to specify the location of the optical axis. If this keyword is omitted, it defaults to the SKY_FROM value.
- OPTAXIS[XY] - The offset of the optical axis from the center of the coordinate system specified by OPTCOORD. The units are the same as the pixel scale (i.e. *xxx*_UNIT) for those coordinates.

3. Examples

Examples of teldef files and plots of the coordinate systems they describe can be found on the Swift Data Center Website at http://swift.gsfc.nasa.gov/sdc/teldef_doc/.. The plots were made using the "draw_teldef" FTOOL. The ASCA teldef files have a number of extra keywords and special defaults and are not particularly good examples of the mission-independent teldef file format. The comments in the Astro-E teldef file may not accurately describe the teldef file format. The files and plots included on the Web are:

Mission	Instrument	Teldef	Plot
Swift	UVOT	sw_uvot_2003-06-02_points.teldef	xis0.gif
	XRT	sw_xrt_2003-02-04.teldef	xrt.gif
Astro-E	XIS	xis0_teldef_1999-12-20.fits	xis0.gif
	XRS	xrs_teldef_1999-12-20.fits	xrs_large.gif / xrs_small.gif
ASCA	SIS	s0_teldef_ascalin.fits	sis0.gif

	GIS	gis2_ano_on_flf_180295.fits	gis2.gif

Appendix 2: Old and New teldef files

(APPENDED TO DEAN HINSHAW'S **"CONVENTIONS FOR ATTITUDE, ALIGNMENT AND COORDINATE TRANSFORMATIONS FOR USE IN SWIFT ANALYSIS SOFTWARE" BY P. BOYD APRIL 22, 2004)**

After the original Swift teldef files were defined, it was pointed out that the Swift spacecraft pointing direction is the +X direction. Additionally, positive spacecraft roll is defined for Swift in the opposite direction of other spacecraft. Neither of these features were represented in the teldef files.

For Swift, the spacecraft Right Ascension, Declination, Roll is defined as an Euler axis sequence from the celestial frame starting with a positive Z axis rotation, followed by a negative Y axis rotation, followed by a negative X axis rotation. Note the negative roll rotation about the boresight (+X) axis.

The attitude and coordinate transformation software assumes +Z is the spacecraft boresight axis, and as a result, the quaterions being written in the pipeline, based on spacecraft RA, Dec and Roll, were referenced to the wrong coordinate system. A solution to this problem was proposed in the form of a new, spacecraft-level teldef file to represent the coordinate transformation between (in the case of Swift) a +X-boresight-oriented spacecraft coordinate system to the +Z-oriented software coordinate system. The new file is called the spacecraft alignment TELDEF file, and the current version in the pipeline is named sw_sc_2004-01-02.teldef. To account for the roll convention, the spacecraft alignment TELDEF file also includes two option keywords:

*ROLLSIGN - Gives the sign of the roll angle, either -1 or +1. Defaults to +1.
*ROLLOFF - Gives the offset of the roll angle in degrees. Defaults to 0.

A positive roll angle rotates the detector coordinates clockwise with respect to the sky coordinates.

The spacecraft alignment TELDEF file gives the SDC software the correct alignment information. The result is that the pipeline will write the quaternions in the correct (spacecraft-oriented) coordinate system.

Changes required.

The changes needed are : a new file, the spacecraft teldef file, and a reordering of the matrix elements in the instrument teldef files.

1) spacecraft TELDEF file with the roll angle. SATi are the spacecraft axes and BOREj are the software boresite axes. The matrix Aij then represents the transformation from the spacecraft to the software coordinates:
   SATi=Σ (Aij*BOREj), where

$$\text{Aij=} \quad \begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}$$

Aij (*i* represents row number in the 3x3 matrix and *j* is the column number) is also known as the "alignment matrix".

2) A new template for the instrument level TELDEF files, which reshuffles the old M_{ij} in the instrument teldef file. None of the instrument teldef files have been updated for the attitude recoding. The new matrix, Nij, will be represented in the ALIGNM*nm* keywords. The new instrument matrix represents the transformation from the spacecraft coordinate system to the detector coordinate system:
   SATi=Σ (Nij*DETj), where, in the case of perfect instrument boresite alignment:

$$\text{Nij=} \quad \begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}$$

In general, to go from old style instrument teldef matrix elements to the new style, one simply swaps entire rows of matrix elements as shown below:

N11= M31
N12= M32
N13= M33
N21= M11
N22= M12
N23= M13
N31= M21
N32= M22
N33= M23

Before the software change, these matrices had the following elements:

Spacecraft (Aij)			Instruments (perfect alignment example) (Mij)		
1	0	0	1	0	0

0	1	0	0	1	0
0	0	1	0	0	1
(or absent)					

After the software change these matrices will instead look like this:

Spacecraft (Aij)			Instruments (Nij)					
0	0	1	0	0	1	M31	M32	M33
1	0	0	1	0	0	M11	M12	M13
0	1	0	0	1	0	M21	M22	M23

In other words, both sets of alignment matrix elements, for the instrument and the observatory, are oriented the same way, because they both represent alignments w/r/t the spacecraft +X axis.

Examples of the new files expected, compared with old files

Old Spacecraft Alignment File

This section includes excerpts from the spacecraft and instrument teldef fits files used in the current pipeline, and drafts planned in the next pipeline, which will accurately handle spacecraft pointing information.

Below is the old spacecraft alignment teldef file, 'sw_sc_2003-10-27.teldef', which is in the current (2004-04-15) pipeline, along with the “oldstyle” instrument teldef files.

```
TELESCOP= 'SWIFT   '           / Telescope (mission) name
INSTRUME= 'SC      '           / Instrument name
FILENAME= 'sw_sc_2003-10-27.teldef' / Suggested file name
DATE    = '2003-10-27'         / FITS file creation date (yyyy-mm-dd)
COMMENT --------------------------------------------------------------------
COMMENT CALDB keywords
COMMENT --------------------------------------------------------------------
CCLS0001= 'BCF     '           / Basic Calibration File
CCNM0001= 'TELDEF  '           / Type of calibration data
CDTP0001= 'DATA    '           / real data, not subroutine
CVSD0001= '2002-10-01'         / Validity start date
CVST0001= '00:00:00'           / Validity start time
CDES0001= 'TELESCOPE DEFINITION FILE' / Brief descriptive summary of this file
COMMENT --------------------------------------------------------------------
COMMENT This alignment matrix specifies how the spacecraft coordinate system
COMMENT and the boresight coordinate system are oriented with respect to each
COMMENT other
COMMENT --------------------------------------------------------------------
ALIGNM11=                  1.0 / DET -> SAT coordinates alignment matrix Mij
```

```
ALIGNM12=            0.0
ALIGNM13=            0.0
ALIGNM21=            0.0 / [3x3 rotation matrix, common to all sensors]
ALIGNM22=            1.0
ALIGNM23=            0.0 / SATX =    M11*DETX + M12*DETY + M13*DETZ
ALIGNM31=            0.0 / SATY =    M21*DETX + M22*DETY + M23*DETZ
ALIGNM32=            0.0 / SATZ =    M31*DETX + M32*DETY + M33*DETZ
ALIGNM33=            1.0
HISTORY ----------------------------------------------------------------------
HISTORY 2004-01-02  Rough draft by Dean Hinshaw
HISTORY            Unit matrix
```

New Spacecraft Alignment File

Below is the way the new spacecraft alignment teldef file, proposed name 'sw_sc_2004-07-15_x.teldef', should look. An “_x” is placed before the .teldef extension to easily distinguish the x-oriented files from the old style files. Note that in addition to the reordered matrix elements, keywords tracking the roll positive direction are included:

```
TELESCOP= 'SWIFT   '           / Telescope (mission) name
INSTRUME= 'SC      '           / Instrument name
FILENAME= 'sw_sc_2003-10-27.teldef' / Suggested file name
DATE    = '2003-10-27'         / FITS file creation date (yyyy-mm-dd)
COMMENT ----------------------------------------------------------------------
COMMENT CALDB keywords
COMMENT ----------------------------------------------------------------------
CCLS0001= 'BCF     '           / Basic Calibration File
CCNM0001= 'TELDEF  '           / Type of calibration data
CDTP0001= 'DATA    '           / real data, not subroutine
CVSD0001= '2002-10-01'         / Validity start date
CVST0001= '00:00:00'           / Validity start time
CDES0001= 'TELESCOPE DEFINITION FILE' / Brief descriptive summary of this file
COMMENT ----------------------------------------------------------------------
COMMENT This alignment matrix specifies how the spacecraft coordinate system
COMMENT and the boresight coordinate system are oriented with respect to each
COMMENT other
COMMENT ----------------------------------------------------------------------
ALIGNM11=            0.0 / DET -> SAT coordinates alignment matrix Mij
ALIGNM12=            0.0
ALIGNM13=            1.0
ALIGNM21=            1.0 / [3x3 rotation matrix, common to all sensors]
ALIGNM22=            0.0
ALIGNM23=            0.0 / SATX =    M11*DETX + M12*DETY + M13*DETZ
ALIGNM31=            0.0 / SATY =    M21*DETX + M22*DETY + M23*DETZ
ALIGNM32=            1.0 / SATZ =    M31*DETX + M32*DETY + M33*DETZ
ALIGNM33=            0.0
```

```
COMMENT ----------------------------------------------------------------------
COMMENT The roll convention keywords below specific the direction in which
COMMENT positive roll increases.
COMMENT ROLLSIGN gives the sign of the roll angle, either -1 or +1.
COMMENT ROLLOFF gives the offset of the roll angle in degrees. Defaults to 0.
COMMENT A positive roll angle rotates the detector coordinates clockwise with respect to
COMMENT the sky coordinates.
COMMENT ----------------------------------------------------------------------
ROLLSIGN=               -1.0 / Sign of roll angle positive direction about boresite axis
ROLLOFF =                0.0 / Offset of roll angle, in degrees
HISTORY ----------------------------------------------------------------------
HISTORY 2004-01-02  Rough draft by Dean Hinshaw
HISTORY             Unit matrix
HISTORY 2004-04-14  New alignment matrix rough draft by Padi Boyd
HISTORY             Describes transformation between spacecraft and software coordinates
HISTORY       Spacecraft is boresite +X, while software assumes boresite is +Z
HISTORY        Also records correct positive roll sign convention.
```

Old BAT Instrument teldef File

Below is the BAT instrument teldef file, ‘sw_bat_2003-02-23.teldef’, which is running in the current SDC pipeline. The instrument alignment matrix is represented as the unit matrix at this time.

```
COMMENT The orientation of the DET coordinates with respect to the
COMMENT satellite axes is specified by the following matrix:
COMMENT ----------------------------------------------------------------------
ALIGNM11=               1.0 / DET -> SAT coordinates alignment matrix Mij
ALIGNM12=               0.0
ALIGNM13=               0.0
ALIGNM21=               0.0 / [3x3 rotation matrix, common to all sensors]
ALIGNM22=               1.0
ALIGNM23=               0.0 / SATX =    M11*DETX + M12*DETY + M13*DETZ
ALIGNM31=               0.0 / SATY =    M21*DETX + M22*DETY + M23*DETZ
ALIGNM32=               0.0 / SATZ =    M31*DETX + M32*DETY + M33*DETZ
ALIGNM33=               1.0
COMMENT ----------------------------------------------------------------------
COMMENT The plate scale is determined from the size of the SKY pixels in the
COMMENT focal plane and the focal length of the telescope.
COMMENT 1 mm roughly corresponds to atan(1/FOCALLEN) radians on tke sky.
COMMENT ----------------------------------------------------------------------
FOCALLEN=             999.5 / Telescope focal length (mm)
COMMENT ----------------------------------------------------------------------
COMMENT  The true optical axis position is not used in the coordinate
COMMENT transformations, but is needed to calculate the detector response.
COMMENT ----------------------------------------------------------------------
```

```
OPTAXISX=            142.5 / optical axis x in DET coordinates (pixel)
OPTAXISY=             87.0 / optical axis y in DET coordinates (pixel)
HISTORY -------------------------------------------------------------------
HISTORY 2003-02-23  Rough draft by C. Markwardt craigm@lheamail.gsfc.nasa.gov
HISTORY             Values are *not* definitive.
```

New BAT Instrument teldef File

Below is the new style BAT instrument teldef file, proposed name 'sw_bat_2004-07-15_x.teldef', which will be placed in the next SDC pipeline to account for the correct direction of spacecraft pointing. An "_x" is placed before the .teldef extension to easily distinguish the x-oriented files from the old style files. The instrument alignment matrix elements have been reorganized according to the scheme described earlier. Since the roll convention keywords appear in the spacecraft alignment teldef file, they are not necessary to include at the instrument level.

```
COMMENT The orientation of the DET coordinates with respect to the
COMMENT satellite axes is specified by the following matrix:
COMMENT -------------------------------------------------------------------
ALIGNM11=             0.0 / DET -> SAT coordinates alignment matrix Mij
ALIGNM12=             0.0
ALIGNM13=             1.0
ALIGNM21=             1.0 / [3x3 rotation matrix, common to all sensors]
ALIGNM22=             0.0
ALIGNM23=             0.0 / SATX =   M11*DETX + M12*DETY + M13*DETZ
ALIGNM31=             0.0 / SATY =   M21*DETX + M22*DETY + M23*DETZ
ALIGNM32=             1.0 / SATZ =   M31*DETX + M32*DETY + M33*DETZ
ALIGNM33=             0.0
COMMENT -------------------------------------------------------------------
COMMENT The plate scale is determined from the size of the SKY pixels in the
COMMENT focal plane and the focal length of the telescope.
COMMENT 1 mm roughly corresponds to atan(1/FOCALLEN) radians on tke sky.
COMMENT -------------------------------------------------------------------
FOCALLEN=            999.5 / Telescope focal length (mm)
COMMENT -------------------------------------------------------------------
COMMENT  The true optical axis position is not used in the coordinate
COMMENT transformations, but is needed to calculate the detector response.
COMMENT -------------------------------------------------------------------
OPTAXISX=            142.5 / optical axis x in DET coordinates (pixel)
OPTAXISY=             87.0 / optical axis y in DET coordinates (pixel)
HISTORY -------------------------------------------------------------------
HISTORY 2003-02-23  Rough draft by C. Markwardt craigm@lheamail.gsfc.nasa.gov
HISTORY             Values are *not* definitive.
HISTORY 2004-04-14  Ammended by Padi Boyd
HISTORY             to describe transformation between spacecraft and BAT coordinates
HISTORY       Spacecraft is boresite +X, while software assumes boresite is +Z
```

While it has not been decided if the teldef file name will carry two extra characters to easily distinguish old-style teldef files from the new ones, a new file name structure such as "sw_bat_2004-07-15_x.teldef", for example, would make clear from the "_x" the the files represent orientation w/r/t +X.

Changes required by software interfaces:

The following software interfaces will be affected by this change.

1) telemetry to FITS conversion (swift2fits). In order to write the correctly oriented quaterions, swift2fits will need to read in the new spacecraft alignment teldef file.
2) Quaterion interpretation routines. Tasks such as attdump and prefilter, which need to translate quaterions into proper values of RA, Dec and Roll, will need to read and interpret the new spacecraft alignment teldef file to properly perform the translation
3) Sky coordinate transformations. Tasks such as batfftimage and swiftxform, which perform transformations between raw and detector coordinates to sky coordinates will still use the instrument teldef files to perform the appropriate transformations. While the interface will not change here, the teldef file will be the "new style" to reflect the change in coordinate system.

Related information:

Complete documentation on teldef files is available at Swift Data Center Website: http://swift.gsfc.nasa.gov/sdc/teldef_doc/.

Note: the reported spacecraft RA and Dec are intended to be the pointing direction of the XRT boresite. After in-flight boresite calibration, offset coordinates will be uploaded so that the ACS data received on the ground is actually reporting coordinates w/r/t XRT. The Telescope Alignment Monitor (TAM) measures and reports the instantaneous misalignment between XRT boresight and star tracker boresight.