Juno
Jovian Auroral Distributions Experiment

JADE Standard Product
Data Record and Archive Volume
Software Interface Specification

Version 04.00
2023-April-30

Prepared by

R. J. Wilson

Juno
Jovian Auroral Distributions Experiment

JADE Standard Product
Data Record and Archive Volume
Software Interface Specification

Custodian:

Rob J. Wilson JADE Archivist	Date

Approved:

Frédéric Allegrini JADE Lead	Date
William Kurth Juno Archivist	Date
Prachet Mokashi JSOC Manager	Date
Raymond J. Walker PDS/PPI Node Manager	Date
Reta Beebe PDS/ATMOS Node Manager	Date

Table of Contents

List of Figures

List of Tables

1 Introduction

This software interface specification (SIS) describes the format and content of the Jovian Auroral Distributions Experiment (JADE) Planetary Data System (PDS) data archive. It includes descriptions of the Standard Data Products and associated metadata, and the volume archive format, content, and generation pipeline.

1.1 Distribution list

Table 1: Distribution list

Name	Organization	Email
Rob Wilson	LASP, JADE archivist	rob.wilson@lasp.colorado.edu
John Hanley	SwRI, JADE FSW specialist	jhanley@swri.edu
Frédéric Allegrini	SwRI, JADE Lead and JADE-E Scientist	fallegrini@swri.edu
Rob Ebert	SwRI, JADE-I Scientist	rebert@swri.edu
Phil Valek	SwRI	PValek@swri.edu
Bradley Trantham	SwRI	BTrantham@swri.edu
Fran Bagenal	LASP	Fran.Bagenal@lasp.colorado.edu
Eric Fattig	SwRI, JADE Operations Lead	eric.fattig@swri.org
Judy Furman	JADE Operations Second	judy.furman@swri.org
Jennifer R. Gale	SwRI, JADE Project Manager	jennifer.gale@swri.org
David McComas	Princeton	DMcComas@princeton.edu
Prachet Mokashi	JSOC Manager	PMokashi@swri.edu
William Kurth	Juno Archivist	william-kurth@uiowa.edu
Reta Beebe	PDS/ATMOS Node Manager	rbeebe@nmsu.edu
Steve Joy	UCLA/PDS/PPI	sjoy@igpp.ucla.edu
Joe Mafi	UCLA/PDS/PPI	jmafi@igpp.ucla.edu
Ray Walker	UCLA/PDS/PPI	rwalker@igpp.ucla.edu

1.2 Document change log

Table 2: Document change log

Change	Date	Affected portion
Initial template	01/15/2010	All
First draft for JADE	04/15/2013	All
Second draft for JADE	08/30/2013	All
FSW 3 Level 3 SIS parts added	06/30/2014	DPID changed to DPID_COUNT Removed Bill Knopf, Michael New and Tom Morgan from the signature list (email 20th Nov '13) Version 02 Level 2 data: Change is an extra object (SCLKSCET_VERSION), and a bug in JAD_HRS_ION_TOF has been fixed from V01. Only V02 now provided to PDS.
FSW3 Level 2 Liens addressed. (Since accepted.)	06/30/2015	All
First draft of FSW4 (rather than FSW3 above) SIS.	12/13/2015	All previous SIS documents were for FSW 3, 2011-2014 data. In 2015 FSW 4 was uploaded, with new formats, and a new PDS Volume was started. This is the first draft of a FSW 4 only SIS. Previous FSW 3 will have its own volume in the PDS
Second draft of FSW4	11/07/2016	All. Updates to SIS for PDS peer review for Cruise Data Delivery deadline.
Third draft of FSW4	01/04/2017	Clarifying updates here and there while waiting for PDS peer review. Bit 4 and Bit 12 added to ISSUES flag in Table 45, and LRS/CAL ion species when ACCUMULATIONT_TIME = 30 warnings added to section 6.2.9.2.3.
Fourth draft of FSW4 to meet PDS peer review liens	05/19/2017	Clarifications and extra explanations, etc. in response to PDS peer review liens. Section 'Occasional jitter in reported times' added.
Version 01 released	07/17/2017	All – PDS peer review completed
Version 02 released	08/07/2018	An update for L3 V02 files. For differences from SIS Version 01 to SIS Versions 02: see JADE_FSW4_SIS_V02_DIFF_V01.PDF
Version 03 released	11/01/2021	An update for L3 electron files V03 (no V03 ion files), and L3 V04 files (both electron and ion). For differences from SIS Version 02 to SIS Versions 03: see JADE_FSW4_SIS_V03_DIFF_V02.PDF
Version 04 released	04/30/2023	Added L5 sections. Renamed INSTRUMENT_NAME field in LBL files to JOVIAN AURORAL DISTRIBUTIONS EXPERIMENT (instead of JOVIAN AURORAL PLASMA DISTRIBUTIONS EXPERIMENT). For differences from SIS Version 03 to SIS Versions 04: see JADE_FSW4_SIS_V04_DIFF_V03.PDF

1.3 TBD items

Table 3 lists items that are not yet finalized.

Table 3: List of TBD items

Item	Sections	Pages(s)

1.4 Abbreviations

Table 4: Abbreviations and their meaning

Abbreviation	Meaning
ASCII	American Standard Code for Information Interchange
BLOB	Binary Large OBject, i.e. a data blob is a large array of binary data
BRT	JADE Burst mode (FSW 3 only)
CAL	JADE Calibration mode
CATS	Juno version of CATS (Cassini Archive Tracking System)
CCSDS	Consultative Committee for Space Data Systems
CD-ROM	Compact Disc – Read-Only Memory
CDR	Calibrated Data Record
CFDP	CCSDS File Delivery Protocol
CHAR	Bytes representing a character string
CK	C-matrix Kernel (NAIF orientation data)
CNT	Units of counts per second, used in filenames
CODMAC	Committee on Data Management, Archiving, and Computing
CRC	Cyclic Redundancy Check
DAC	Digital to Analogue Conversion
DAP	Data Analysis Product
DAT	PDS binary file
DDR	Derived Data Record
DER	JADE Direct Events (Raw) mode
DES	JADE Direct Events (Split-out) mode
DOUBLE	An 8-byte (double-precision) real floating point value
DMAS	Data Management and Storage
DSN	Deep Space Network
DVD	Digital Versatile Disc
DVD-R	DVD – Recordable media
E&PO	Educational and Public Outreach
EDA	End of data acquisition
EDR	Experiment Data Record
EFB	Earth Fly By
EGA	Earth Gravity Assist
ELC	ELeCtron sensor
SPDR	Standard Product (Experiment and Pipeline) Data Record
FEI	File Exchange Interface
FGM	The 3-letter code for the magnetometer instrument on Juno. Also known as MAG.
FLOAT	A 4-byte (single-precision) real floating point value
FMT	PDS Format file
FOV	Field of View
FSW	Flight Software

FTP	File Transfer Protocol
GB	Gigabyte(s)
GCR	Galactic Cosmic Ray
GRAV	Orbit type focusing on gravity rather than MWR
GSFC	Goddard Space Flight Center
HK	Housekeeping
HLC	High rate, Low rate and Calibration mode
HLS	High and Low rate Science (or HLC minus Calibration)
HRS	JADE High Rate Science mode
HSK	JADE Housekeeping
HTML	Hypertext Markup Language
HV	High Voltage
HVE	High Voltage Engineering mode
HVENG	High Voltage ENGineering mode
HVCO	High Voltage Check Out
ICD	Interface Control Document
INT8	8-bit (1-byte) Signed Integer
INT16	16-bit (2-bytes) Signed Integer
INT32	32-bit (4-bytes) Signed Integer
IOT	Instrument Operations Team
ISO	International Standards Organization
JADE	Jovian Auroral Distributions Experiment
JEDI	Jupiter Energetic Particle Detector Instrument
JIRAM	Jupiter InfraRed Auroral Mapper
JOI	Jupiter Orbit Insertion
JPL	Jet Propulsion Laboratory
JSC	Johnson Spaceflight Center
JSOC	Juno Science Operations Center
JSS	Jupiter De-Spun-Sun co-ordinate system
LASP	Laboratory for Atmospheric and Space Physics, University of Colorado
LBL	PDS label file
LET	Lineal Energy Transport
LSB	Least Significant Byte first (also known as little endian)
LSB_INTEGER	PDS binary format: 1-, 2-, and 4- byte signed integers (little endian)
LSB_UNSIGNED_INTEGER	PDS binary format: 1-, 2-, and 4- byte unsigned integers (little endian)
LOG	LOGigical counters (ion data)
LRS	JADE Low Rate Science mode
LUT	Look-Up Table(s)
MAG	Magnetometer Instrument (also known as FGM)
MB	Megabyte(s)
MCP	Micro Channel Plate

MOS	Mission Operations System
MWR	Microwave Radiometer Instrument
NAIF	Navigation and Ancillary Information Facility (JPL)
NASA	National Aeronautics and Space Administration
NSSDC	National Space Science Data Center
ODL	Object Description Language
PCK	Planetary Cartographic and Physical Constants Kernel (NAIF)
PC_REAL	PDS binary format: 4-, 8-, and 10- byte real numbers in IBM/PC format (little endian)
PDS	Planetary Data System
PPI	Planetary Plasma Interactions Node (PDS)
RDR	Reduced Data Record
RPM	Revolutions per Minute
RSSG	Radio Science System Group
SCET	Spacecraft Event Time
SCLK	Spacecraft Clock
SEU	Single Event Upset
SIS	Software Interface Specification
SOC	Science Operations Center
SPDR	Standard Product Data Record
SPE	Solar Particle Event
SPICE	Spacecraft, Planet, Instrument, C-matrix, and Events, (NAIF data format)
SPWG	Science Planning Working Group
SPK	SPICE (ephemeris) Kernel (NAIF)
SSH	Secure Shell
SwRI	Southwest Research Institute
TAR	Tape Archives (file format)
TBC	To Be Confirmed
TBD	To Be Determined
TEP	Tissue Equivalent Plastic
TOF	Time Of Flight
UINT8	8-bit (1-byte) Unsigned Integer
UINT16	16-bit (2-bytes) Unsigned Integer
UINT32	32-bit (4-bytes) Unsigned Integer
UCLA	University of California, Los Angeles
UVS	Ultraviolet Spectrometer Instrument
V-EGA	Venus-Earth Gravity Assist

1.5 Glossary

Archive – An archive consists of one or more data sets along with all the documentation and ancillary information needed to understand and use the data. An archive is a logical construct independent of the medium on which it is stored.

Archive Volume – A volume is a logical organization of directories and files in which data products are stored. An *archive volume* is a volume containing all or part of an archive; i.e. data products plus documentation and ancillary files.

Archive Volume Set – When an archive spans multiple volumes, they are called an *archive volume set*. Usually the documentation and some ancillary files are repeated on each volume of the set, so that a single volume can be used alone.

Catalog Information – High-level descriptive information about a data set (e.g. mission description, spacecraft description, instrument description), expressed in Object Description Language (ODL), which is suitable for loading into a PDS catalog.

CODMAC Levels – Descriptive data labels to inform you of the amount of processing from the original raw data product (as defined by the Committee on Data Management, Archiving, and Computing). These are different to NASA levels. Note that JADE data does not require CODMAC level 4 files; the PDS will ultimately contains JADE Level 2, 3 and 5 data. See Table 5 for the different level definitions.

Data Product – A labeled grouping of data resulting from a scientific observation, usually stored in one file. A product label identifies, describes, and defines the structure of the data. An example of a data product is a planetary image, a spectral table, or a time series table.

Data Set – A data set is an accumulation of data products together with supporting documentation and ancillary files.

Experiment Data Record – An accumulation of raw output data from a science instrument, in chronological order, with duplicate records removed, together with supporting documentation and ancillary files.

Pipeline Data Record – An accumulation of calibrated data from a science instrument, derived from experiment data records, together with supporting documentation, calibration data, and ancillary files.

Standard Data Product – A data product generated in a predefined way using well-understood procedures and processed in "pipeline" fashion. Data products that are generated in a non-standard way are sometimes called *special data products*.

Table 5: CODMAC Levels of Data Descriptions and Meaning to JADE

CODMAC Level	Description	Meaning for JADE	Stored JSOC / PDS	NASA Level
1 (Raw)	Telemetry data stream as received at the ground station, with science and engineering data embedded.	Packet Data (Highly Compressed)	No	Packet Data
2 (Edited)	Instrument science data (e.g., raw voltages, counts) at full resolution, time ordered, with duplicates and transmission errors removed.	Unpacked Data (Engineering Units.)	Yes	0
3 (Calibrated)	Level 2 data that have been located in space and may have been transformed (e.g., calibrated, rearranged) in a reversible manner and packaged with needed ancillary-data (e.g., radiances with the calibration equations applied).	Unpacked Data (Scientific Units.)	Yes	1-A
4 (Resampled)	Irreversibly transformed (e.g., resampled, remapped, calibrated) values of the instrument measurements (e.g., radiances, magnetic field strength).	Not used for JADE (Better for images than plasma data)	N/A	1-B
5 (Derived)	Level 3 or 4 data that have been resampled and mapped onto uniform space-time grids. The data are calibrated (i.e., radiometrically corrected) and may have additional corrections applied (e.g., terrain correction).	e.g. Plasma Parameters or Moments Pitch Angle Distributions	Yes	1-C
	Geophysical parameters, generally derived from Level 3 or 4 data, and located in space and time commensurate with instrument location, pointing, and sampling.			2
	Geophysical parameters mapped onto uniform Space-time grids.			3
	Any product that also requires data from another instruments for its derivation. (i.e. electron pitch angle derivation requires both level 3 MAG and level 3 electron data.)			2

Table inspired by Appendix F of the Planetary Data System Archive Preparations Guide (APG), Version 1.4 (April 1, 2010) found at https://pds.nasa.gov/documents/apg/apg.pdf.
However they compared CODMAC Levels to NASA Levels and had a description based on NASA levels, which this author has altered in the table above to refer to CODMAC levels. This author also added the final level 5 line about products requiring additional data from other instruments.

1.6 Juno Mission Overview

Juno launched on the first day of its launch window, 5 August 2011. The spacecraft used a ΔV-EGA trajectory consisting of deep space maneuvers on 08 August 2012 and 14 September 2012 followed by an Earth gravity assist (EGA) on 9 October 2013. Jupiter arrived on 5 July 2016 (UTC), using two 53-day capture orbits prior to commencing operations for a 5-(Earth) year long prime mission comprising 34 high inclination, high eccentricity orbits of Jupiter. Instead of firing the engines a second time to get to the originally intended 14-day orbits, it was decided not to, and remain in the 53-day orbital periods (altering the 34 orbit prime mission duration from the original 1 year to 5 years). The orbit is polar (90° inclination) with a periapsis altitude of ~4200 km and a semi-major axis of ~113 R_J (1 R_J is one Jovian radius, ~71492 km). The primary science is acquired for ~6 hours, ~centered on each periapsis although fields and particles data are acquired at low rates for the remaining orbit. Of the first 9 periapses, 4 were dedicated to microwave radiometry (MWR orbits) of Jupiter's deep atmosphere, 4 were dedicated to gravity measurements (GRAV orbits) to determine the structure of Jupiter's interior, and Juno went in to Safe mode on orbit 2 resulting in no perijove data. All orbits will include fields and particles measurements of the planet's auroral regions. Juno is spin stabilized with a rotation rate of 1 to 3 revolutions per minute (RPM). For the MWR orbits the spin axis is, usually, perpendicular to the orbit plane so that the radiometer fields of view pass through the nadir, but is tilted for some orbits. For gravity passes, the spin axis is aligned to the Earth direction, allowing for Doppler measurements through the periapsis portion of the orbit. The orbit plane is initially very close to perpendicular to the Sun-Jupiter line and evolves over the mission. Data acquired during the periapsis passes are recorded and played back over the subsequent apoapsis portion of the orbit.

Juno's instrument complement includes Gravity Science using the X and Ka bands to determine the structure of Jupiter's interior; vector fluxgate magnetometer (MAG) to study the magnetic dynamo and interior of Jupiter as well as to explore the polar magnetosphere; and a microwave radiometer (MWR) experiment covering 6 wavelengths between 1.3 and 50 cm to perform deep atmospheric sounding and composition measurements. The instrument complement also includes a suite of fields and particle instruments to study the polar magnetosphere and Jupiter's aurora. This suite includes an energetic particle detector (JEDI), a Jovian auroral (plasma) distributions experiment (JADE), a radio and plasma wave instrument (Waves), an ultraviolet spectrometer (UVS), and a Jupiter infrared auroral mapping instrument (JIRAM). The JunoCam is a camera included for education and public outreach. While this is not a science instrument, we plan to capture the data and archive them in the PDS along with the other mission data. Appendix A includes Lead Co-Is and archivists for JADE, along with the associated PDS Discipline Node.

1.7 SIS Content Overview

Section 2 describes the JADE instrument. Section 3 describes the data sets, data flow, and validation. Section 4 describes the structure of the archive volumes and contents of each file. Section 5 describes the file formats used in the archive volumes.
Individuals responsible for generating the archive volumes are listed in Appendix A. PDS-compliant label files for all JADE standard data products are itemized and described in Appendix B, while the data products file headers and data record formats are itemized and described in section 6, Appendix C, and Appendix D respectively.

1.8 Scope of this document

The specifications in this SIS apply to all JADE Standard Data Record products submitted for archive to the Planetary Data System (PDS), for all phases of the Juno mission. Some sections of this document describe parts of the JADE archive and archiving process that are managed by the PDS archive team. These sections have been provided for completeness of information and are not maintained by the JADE team.

This document is not intended as a JADE Users Guide; it describes the data, not how to interpret the data for science. Seek guidance from the JADE team for how to use the data.

1.9 Applicable Documents

ISO 9660-1988, Information Processing—Volume and File Structure of CD-ROM for Information Exchange, 04/15/1988.

Planetary Data System Archive Preparation Guide, Version 1.1, JPL D-31224, 08/29/2006.

Planetary Data System Standards Reference, JPL D-7669, Part 2, Version 3.8, 02/27/2009.

Planetary Science Data Dictionary Document, Planetary Data System, JPL D-7116, Version 1r65, 02/2007.

Juno Mission Operations Concept Document, JPL D-35531, Version Preliminary, 04/30/2007.
Juno Science Data Management and Archive Plan, Version Final, JPL D-34032, 08/26/2009.

The JADE Instrument Paper (also see section 2):

McComas, D.J. and Alexander, N. and Allegrini, F. and Bagenal, F. and Beebe, C. and Clark, G. and Crary, F. and Desai, M.I. and De Los Santos, A. and Demkee, D. and Dickinson, J. and Everett, D. and Finley, T. and Gribanova, A. and Hill, R. and Johnson, J. and Kofoed, C. and Loeffler, C. and Louarn, P. and Maple, M. and Mills, W. and Pollock, C. and Reno, M. and Rodriguez, B. and Rouzaud, J. and Santos-Costa, D. and Valek, P. and Weidner, S. and Wilson, P. and Wilson, R.J. and White, D. (2017), The Jovian Auroral Distributions Experiment (JADE) on the Juno Mission to Jupiter, *Space Science Reviews*, **213**, 547-643, doi: 10.1007/s11214-013-9990-9

1.10 Audience

This document is useful to those wishing to understand the format and content of the JADE PDS data product archive collection. Typically, these individuals would include scientists, data analysts, or software engineers.

2 JADE Instrument Description

Rather than repeat information, we refer the reader to the **Open Access** instrument paper in Space Science Reviews (SSR) for a full description of the JADE instrument. Below we provide the DOI link to the paper, reference and the abstract that gives an overview of the instrument.

NOTE: JADE had FSW 3 when this instrument paper was written and published. Since then we use FSW 4 and the JADE products are different. This SIS is the best description of the FSW 4 products, however the actual hardware and science goals remain the same.

Official SSR citation and DOI:
McComas, D.J., Alexander, N., Allegrini, F. et al. Space Sci Rev (2017) 213: 547.
https://doi.org/10.1007/s11214-013-9990-9

AGU style reference:
McComas, D. J., *et al.* (2017), The Jovian Auroral Distributions Experiment (JADE) on the Juno Mission to Jupiter, *Space Science Reviews*, **213**, 547-643, doi:10.1007/s11214-013-9990-9.

The paper was accepted and published online at SSR in 2013, hence some references may have that year. When the Juno special issue came out in 2017, SSR altered the year to 2017 for all Juno instruments papers, but is otherwise the same (same DOI, same paper, only the publish year altered and a printed issue volume and page numbers are now included in the citation).

Abstract:
"The Jovian Auroral Distributions Experiment (JADE) on Juno provides the critical in situ measurements of electrons and ions needed to understand the plasma energy particles and processes that fill the Jovian magnetosphere and ultimately produce its strong aurora. JADE is an instrument suite that includes three essentially identical electron sensors (JADE-Es), a single ion sensor (JADE-I), and a highly capable Electronics Box (EBox) that resides in the Juno Radiation Vault and provides all necessary control, low and high voltages, and computing support for the four sensors. The three JADE-Es are arrayed 120° apart around the Juno spacecraft to measure complete electron distributions from ~0.1 to 100 keV and provide detailed electron pitch-angle distributions at a 1 s cadence, independent of spacecraft spin phase. JADE-I measures ions from ~5 eV to ~50 keV over an instantaneous field of view of 270° x 90° in 4 s and makes observations over all directions in space each 30 s rotation of the Juno spacecraft. JADE-I also provides ion composition measurements from 1 to 50 amu with $m/\Delta m$ ~2.5, which is sufficient to separate the heavy and light ions, as well as O^+ vs. S^+, in the Jovian magnetosphere. All four sensors were extensively tested and calibrated in specialized facilities, ensuring excellent on-orbit observations at Jupiter. This paper documents the JADE design, construction, calibration, and planned science operations, data processing, and data products. Finally, the Appendix describes the Southwest Research Institute [SwRI] electron calibration facility, which was developed and used for all JADE-E calibrations. Collectively, JADE provides remarkably broad and detailed measurements of the Jovian auroral region and magnetospheric plasmas, which will surely revolutionize our understanding of these important and complex regions."

3 Data Set Overview

3.1 Data Sets

The JADE data archive is divided into 5 data sets. Each data set is subdivided into different standard data product types. A basic description of each data set is provided in Table 6. The standard data product types are described in Table 7. A more detailed description of each data set is provided in the sections that follow these two tables.

The standard data product IDs for Level 2 data are a series of *five* three-letter codes (instrument, CODMAC level, telemetry mode, sensor and data type, respectively) separated by an underscore, of the form:

JAD_L20_***aaa***_***bbb***_***ccc***

The products filenames then append that with a date and version number, of the form:

JAD_L20_***aaa***_***bbb***_***ccc***_***yyyyddd***_V***nn***.DAT
JAD_L20_***aaa***_***bbb***_***ccc***_***yyyyddd***_V***nn***.LBL
JAD_L20_***aaa***_***bbb***_***ccc***_V***nn***.FMT

Where:

JAD Instrument, short for JADE
L20 CODMAC **L**evel **2**, JADE internal convention **0** (zero).
aaa Telemetry mode type:
- ALL = All telemetry modes
- CAL = Calibration telemetry mode
- HLC = High rate, low rate and calibration telemetry modes
- HRS = High rate telemetry mode
- *HSK = Housekeeping telemetry mode*
- HVE = High Voltage Engineering telemetry mode
- LRS = Low rate telemetry mode

bbb Sensor type: ALL, ELC or ION
- ELC = electron sensor(s)
- ION = ion sensor
- *ALL = both ion and electron sensors*

ccc Data type:
- ALL = all three electron sensors, or all eight ion species
- ANY = any of the electron sensors, or any ion species
- DER / DES = ion Direct Events Raw / Split-out
- LOG = ion Logical counters
- TOF = ion Time-Of-Flight
- *BHK, BMS, MEM, ERR, SHK = Not for PDS, JADE operations only*
- *OA0, OA1, OA2, OA3 = Not for PDS, JADE operations only*

yyyy 4-digit year
ddd 3-digit day of year
nn 2-digit version number of file

Values in grey italics are JADE operations products not for science and not in PDS, but listed here for completeness to benefit JADE team members. The FMT file is also within the LBL file.

The standard data product IDs for Level 3 data are a similar series of *five* three-letter codes (instrument, CODMAC level, telemetry mode, sensor, data type and unit, respectively) separated by an underscore, of the form:

L30_***aaa***_***bbb***_***ccc***_***uuu***

[The starting "JAD_" of level 2 files has been dropped from level 3 files as a PDS requirement forbids STANDARD_PRODUCT_ID being more than 20 characters, however JAD_ is still used for level 3 filenames.]

The products filenames then append that with a date and version number, and prepend with JAD_, of the form:

JAD_L30_***aaa***_***bbb***_***ccc***_***uuu***_***yyyyddd***_V***nn***.DAT
JAD_L30_***aaa***_***bbb***_***ccc***_***uuu***_***yyyyddd***_V***nn***.LBL
JAD_L30_***aaa***_***bbb***_***ccc***_***uuu***_V***nn***.FMT

Where:

JAD Instrument, short for JADE
L30 CODMAC **L**evel **3**, JADE internal convention **0** (zero).
aaa Telemetry mode type:
CAL = Calibration telemetry mode, Not for PDS, JADE operations only
HLS = High rate and low rate telemetry mode (not including calibration)
HRS = High rate telemetry mode
LRS = Low rate telemetry mode
bbb Sensor type:
ELC = electron sensor(s)
ION = ion sensor
ccc Data type:
ALL = all three electron sensors, or all eight ion species
TWO = Electron sensors E060 and E180, but not E300.
ANY = any of the electron sensors, or any ion species
LOG = ion Logical counters
TOF = ion Time-Of-Flight
uuu Unit type:
CNT = counts per second
DEF = Differential Energy Flux [Not currently used in L3, see L5]
yyyy 4-digit year
ddd 3-digit day of year
nn 2-digit version number of file

Values in grey italics are JADE operations products not for science and not in PDS, but listed here for completeness to benefit JADE team members. The FMT file is also within the LBL file.

The standard data product IDs for Level 5 binary data are a similar series of three-letter codes (CODMAC level, telemetry mode, sensor, data type and unit, respectively) separated by an underscore, of the form:

L50_***aaa***_***bbb***_***ccc***_***uuu***

[The starting "JAD_" of level 2 files has been dropped from level 5 files as a PDS requirement forbids STANDARD_PRODUCT_ID being more than 20 characters, however JAD_ is still used for level 5 filenames.]

The products filenames then append that with a date and version number, and prepend with JAD_, of the form:

JAD_L50_***aaa***_***bbb***_***ccc***_***uuu***_***yyyyddd***_V***nn***.DAT
JAD_L50_***aaa***_***bbb***_***ccc***_***uuu***_***yyyyddd***_V***nn***.LBL
JAD_L50_***aaa***_***bbb***_***ccc***_***uuu***_V***nn***.FMT

Where:

JAD Instrument, short for JADE
L50 CODMAC **L**evel **5**, JADE internal convention **0** (zero).
aaa Telemetry mode type:
- HLS = High rate and low rate telemetry mode (not including calibration)
- HRS = High rate telemetry mode
- LRS = Low rate telemetry mode

bbb Sensor type:
- ELC = electron sensor(s)
- ION = ion sensor

ccc Data type:
- TWO = Electron sensors E060 and E180, but not E300.
- ANY = any of the electron sensors, or any ion species
- TOF = ion Time-Of-Flight

uuu Unit type:
- CNT = counts per second
- DEF = Differential Energy Flux

yyyy 4-digit year
ddd 3-digit day of year
nn 2-digit version number of file

Values in grey italics are JADE operations products not for science and not in PDS, but listed here for completeness to benefit JADE team members. The FMT file is also within the LBL file.

The standard data product IDs for Level 5 ASCII data are a similar series of mostly three-letter codes (data type, 'tropic' type, dimensions and species, respectively) separated by an underscore, of the form:

ccc_ddd_ee_f

[The starting "JAD_L50_***aaa_bbb***_" of level 5 file names has been dropped from level 5 files as a PDS requirement forbids STANDARD_PRODUCT_ID being more than 20 characters, however JAD_L50_***aaa_bbb***_ is still used for level 5 filenames.]
Not that ***ee*** is a two letter code, and ***f*** can be a variable number of characters in length.

The products filenames then append that with a date and version number, and prepend with JAD_, of the form:

JAD_L50_***aaa_bbb_ccc_ddd_ee_f_yyyyddd***_V***nn***.CSV
JAD_L50_***aaa_bbb_ccc_ddd_yyyyddd***_V***nn***.LBL
JAD_L50_***aaa_bbb_ccc_ddd***_V***nn***.FMT

Where:

JAD Instrument, short for JADE
L50 CODMAC **L**evel **5**, JADE internal convention **0** (zero).
aaa Telemetry mode type:
- HLS = High rate and low rate telemetry mode (not including calibration)
- HRS = High rate telemetry mode
- LRS = Low rate telemetry mode

bbb Sensor type:
- ELC = electron sensor(s)
- ION = ion sensor

ccc Data type:
- MOM = Numerical Moments

ddd 'tropic' type:
- ISO = Isotropic plasma distribution assumed
- ANI = Anisotropic plasma distribution assumed

ee Dimensions used:
- 1D = 1 Dimension used for calculations
- 2D = 2 Dimension used for calculations
- 3D = 3 Dimension used for calculations

f Species type:
- ELECTRONS = Electrons
- PROTONS = Protons
- HEAVIES = Heavy Ions (i.e. m/q > 5)

yyyy 4-digit year
ddd 3-digit day of year
nn 2-digit version number of file

Values in grey italics are JADE operations products not for science and not in PDS, but listed here for completeness to benefit JADE team members. The FMT file is also within the LBL file.

Table 6: Relationship Between Data Sets and Standard Data Products

Data Set ID	CODMAC Level	Standard Data Product ID	ID
JNO-SW-JAD-2-UNCALIBRATED-V1.0 Uncalibrated science data 2011 to 2014 inclusive, using FSW 3 data.	2	See FSW 3 SIS document for details, available within that PDS volume.	P0
JNO-J/SW-JAD-2-UNCALIBRATED-V1.0 Uncalibrated science data 2015 onwards, using FSW 4 data.	2	JAD_L20_ALL_ION_DER JAD_L20_ALL_ION_DES JAD_L20_CAL_ELC_ALL JAD_L20_CAL_ION_ANY JAD_L20_HLC_ION_LOG JAD_L20_HLC_ION_TOF JAD_L20_HRS_ELC_ALL JAD_L20_HRS_ION_ANY JAD_L20_HVE_ELC_ALL JAD_L20_HVE_ION_ALL JAD_L20_HVE_ION_LOG JAD_L20_HVE_ION_TOF JAD_L20_LRS_ELC_ANY JAD_L20_LRS_ION_ANY	P0
JNO-J/SW-JAD-3-CALIBRATED-V1.0 Calibrated JADE data 2015 onwards only. All data prior to 2015 (FSW 3) was operational only and contained no science intervals.	3	L30_HLS_ION_LOG_CNT L30_HLS_ION_TOF_CNT L30_HRS_ELC_ALL_CNT L30_HRS_ELC_TWO_CNT L30_HRS_ION_ANY_CNT L30_LRS_ELC_ANY_CNT L30_LRS_ION_ANY_CNT [Note: filenames start JAD_L30_* but no JAD_ in Standard Data Product ID as 20 char limit in PDS.]	P1
		[P2 is merged in with P3, P4, P5]	P2
JNO-J/SW-JAD-5-CALIBRATED-V1.0 Calibrated JADE data, with pitch angles for Electron files, 2015 onwards only.	5	L50_HRS_ION_ANY_DEF L50_LRS_ION_ANY_DEF	P3
		L50_HLS_ION_TOF_DEF	P4
		L50_HRS_ELC_TWO_DEF L50_LRS_ELC_ANY_DEF	P5
JNO-J-JAD-5-MOMENTS-V1.0 plasma moments At Jupiter only, no cruise data.	5	MOM_ISO_3D_HEAVIES MOM_ISO_3D_PROTONS	P6
		MOM_ISO_2D_ELECTRONS	P7

Table 7: Standard Data Product Contents

ID	Key/Physical Parameters	Processing Inputs	Product Format	Description
P0	Reformatted Engineering Data Record (REDR). Time ordered (duplicates removed) full resolution science data (counts at voltage levels). Time ordered counts (per accumulation or per second) vs. voltage level vs. direction.	JADE raw telemetry packets	Binary	Packets are uncompressed, bitmaps expanded to single objects, extra objects added to describe FSW and LUT versions, data units and a UTC timestamp.
P1	Time ordered counts per second in energy vs. look direction, with uncertainties.	JADE P0 data	Binary	Like JADE packets are combined to fewer files with position/auxiliary info necessary to calculate position or moments.
P2	Time ordered {electron or ion} flux vs. direction vs. energy. [Not doing – this data set is merged in with P3, P4 and P5.]	[JADE P1 data]	Binary	[Merged in to P3, P4 and P5]
P3	Time-ordered differential energy fluxes ion species	JADE P1 and MAG data	Binary	The P2 data converted to science units, a magnetic field vector (from MAG) is added.
P4	Time-ordered differential energy fluxes of ion TOF	JADE P1 and MAG data	Binary	The P2 data converted to science units, a magnetic field vector (from MAG) is added.
P5	Time-ordered differential energy fluxes of electron with pitch angles	JADE P1 and MAG data	Binary	The P2 data converted to science units, a magnetic field vector (from MAG) is added, plus pitch angles.
P6	Ion plasma moments (when measurements allow)	JADE P3 and MAG data	ASCII	Ion plasma moments.
P7	Electron plasma moments (when measurements allow)	JADE P5 and MAG data	ASCII	Electron plasma moments.

The following sub-sections describe the different modes of JADE, but are best summarized in the periodic table inspired Figure 6.

3.1.1 MCP Calibration (CAL) Data Set

This is not intended for science use, but for the JADE instrument team to perform calibration tests. It occurs for a total of ~80 minutes per orbit.

The data has a minimum value removed (the MIN_SUBTRACTED_VALUE object), then is compressed from 4-byte values to 1-byte values, then losslessly compressed for transmission. The Direct Events products are the exception in not being compressed in any way.

There is just one electron product that contains all three electron sensors separately but within the same record.

The ion data records are split into ping and pongs, each containing half the energy sweep, which must be merged for Level 3 and higher products.

3.1.2 High Rate Science (HRS) Data Set

This is the highest data rate mode, occurring for a total of ~6 hours per orbit.

The data has a minimum value removed (the MIN_SUBTRACTED_VALUE object), then is compressed from 2-byte values to 1-byte values, then losslessly compressed for transmission. The Direct Events products are the exception in not being compressed in any way.

There is just one electron product that contains all three electron sensors separately but within the same record.

The ion data records are split in to ping and pongs, each containing half the energy sweep, which must be merged for Level 3 and higher products.

3.1.3 High Voltage Engineering (HVE) Data Set

This is not intended for science use, but for the JADE instrument team to perform tests, occurring for a total of ~1 to 2.5 hours per orbit (depending on any maneuvers).

The data is collected as total counts without any compression at all (and does not remove a minimum subtracted value).

There is just one electron product that contains all three electron sensors, and just one ion species product that contains all eight ion species.

The ion data records are split into ping and pongs, each containing half the energy sweep, which must be merged for Level 3 and higher products.

3.1.4 Low Rate Science (LRS) Data Set

This is the most common mode and the lowest data rate, occurring for a total of ~327 hours per orbit, with ~19 hours of that within an intermediate (higher time cadence) LRS mode.

The data has a minimum value removed (the MIN_SUBTRACTED_VALUE object), then is compressed from 4-byte values to 1-byte values, then losslessly compressed for transmission. The Direct Events products are the exception in not being compressed in any way.

There are three electron products, one for each sensor. However, only one electron sensor can be active at a time while in LRS mode.

The ion data records are split into ping and pongs, each containing half the energy sweep, which must be merged for Level 3 and higher products.

3.1.5 Data units for Level 2 Products

The base data unit (for products with PACKETID >10) is total counts for that record (as such are integer numbers), with the exception of JAD_L20_CAL_ION_ANY_*, JAD_L20_LRS_ION_ANY_* and JAD_L20_LRS_ELC_ANY_* products. These are in units of count rates as these are data products related to spin phase and the spin-period may vary slightly. The count rates are total counts divided by number of views within the accumulation time, to the nearest 1/512 of a count (represented as a float).

The number of views is simply the number of times an anode has been included in the record's accumulation time; for instance if two anodes are summed together for a product, then that's two views. If the accumulation is over many spacecraft spin periods and the product is one regarding spin-phase, then every spin the number of views increases for a particular spin-phase angle. The record normalized total counts measured by total number of views, however that is rarely equivalent to units of counts per second.

To be certain you know which units, all products have had an object added to their PDS record, DATA_UNITS, which (for Level 2 data) is either 0 or 1 for total count or rate respectively.

3.1.6 DATA object vs. onboard data BLOB for Level 2 Products

Many JADE products onboard JUNO remove a minimum value from the data prior to compression of the data BLOB for transmission. On the ground when we decompress these data packets we add back on this value so that the end user does not have to; with the MIN_SUBTRACT_VALUE object used reported in the file. If the object did not have a minimum value removed and so not object (e.g. HSK telemetry modes) then for consistency a MIN_SUBTRACT_VALUE object is added to the file on the ground, but if so will be set to zero, and the upper and lower valid limits for that product will be fixed at zero.

3.1.7 Occasional jitter in reported times

Occasionally the reported spacecraft clock value is a second out from where you would expect, e.g. in a series of records all with an ACCUMULATION_TIME of 30 seconds, you may get times that are consistently 30 seconds apart, then (very occasionally) have one that is 29 or 31 seconds apart from its neighbor based on the spacecraft clock (used to make UTC). This is a known Juno feature related to having two spacecraft clocks, but any correction would be a level 3 data product (as it is a time calibration). The two clocks on Juno are not always synchronized, and the time message from them can occasionally have a stutter/jitter where instead of advancing 2 ticks, sometimes it advances 3 ticks, then some 2 ticks, then 1 tick, then returns to the regular 2 tick pattern. JADE Level 2 data records use the onboard reported time message as is, and has no

in-situ way to know if the reported time is during this stutter. However, on the ground we add ISSUES object bit 10 to mark Level 2 records where we know that the JADE packet's TIMESTAMP_WHOLE/SUB has been stuttered. [Note version 01+ files will have this, but version 00 file (not on PDS) are made before the time stutter intervals are known, so may not be flagged until later when we make the non-zero version file.] For Level 3 JADE files, we correct for the Juno time stutter in the TIMESTAMP_WHOLE/SUB object, and adjust the ISSUES object: removing ISSUES bit 10 and flagging ISSUES bit 5.
This Juno time stutter affects all spacecraft clock times reported by JADE, and tends to occur every few days. While we do track the times it affects the JADE packet's TIMESTAMP_WHOLE/SUB, we do not track any others spacecraft clock times. For example the Electron files have MAG_TIMESTAMP_WHOLE/SUB objects which may also be affected, but JADE does not track these nor attempt to correct. This is because publishable work requiring MAG data should be using the MAG team's Version 01+ MAG files, rather than the un-calibrated 'quicklook' spacecraft reported MAG from JADE's files.

For JADE team operations work (files not on PDS), the OSCOPE MET times may also be affected, as could the SPIN_TIME_WHOLE/SUB values in SHK files, neither of which is tracked for time stutter effects. In all cases listed above, the LBL file for these objects note they may be subject to the Juno time stutter.

3.2 Data Flow

The Juno Data Management and Storage (DMAS) will receive packets and CCSDS File Delivery Protocol (CFDP) products from the Deep Space Network (DSN) and place these on the Project data repository system. The DMAS will provide the initial processing of the raw telemetry data bringing it to Committee on Data Management and Archive (CODMAC) Level 1 science data. The JADE Instrument Operations Team (IOT) will retrieve the CODMAC Level 1 data from the DMAS using FEI services and ancillary data from the JPL Mission Support Area (MSA) via Juno Science Operations Center (JSOC). The IOT will decompress the Level 1 data and return them to the JSOC as CODMAC Level 2 data. The JSOC will also receive and organize higher-level data products developed by the Science Investigation Teams associated with each instrument. JSOC development and operations will be carried out at SwRI, in coordination with the MOS at JPL.

The JADE Science Investigation Team will verify the content and the format will be validated. The resulting decompressed, restructured Level 2 data will constitute the lowest level of data to be archived with the PDS. JSOC will coordinate the validation of the edited (CODMAC Level 2) data archive volumes created by the IOT. The Science Investigation Team will develop higher-level data products based on the Level 2 data and ancillary data and return these to the JSOC. JSOC will support archiving the Level 2 data by building archive volumes and verifying the format of the volumes and included data and metadata. Higher-level data set archives will be coordinated through the JSOC. The Science Investigation Team will be responsible for ensuring that the metadata and documentation included with these data sets are complete and accurate. This means that both JSOC and the Science Investigation Team will need to work closely with the PDS. This coordination will be fostered via the Data Archive Working Group.

A comprehensive description of the Juno Mission System is provided in the Juno Mission Operations Concept. A data flow diagram for the downlink process is shown in Fig. 1.

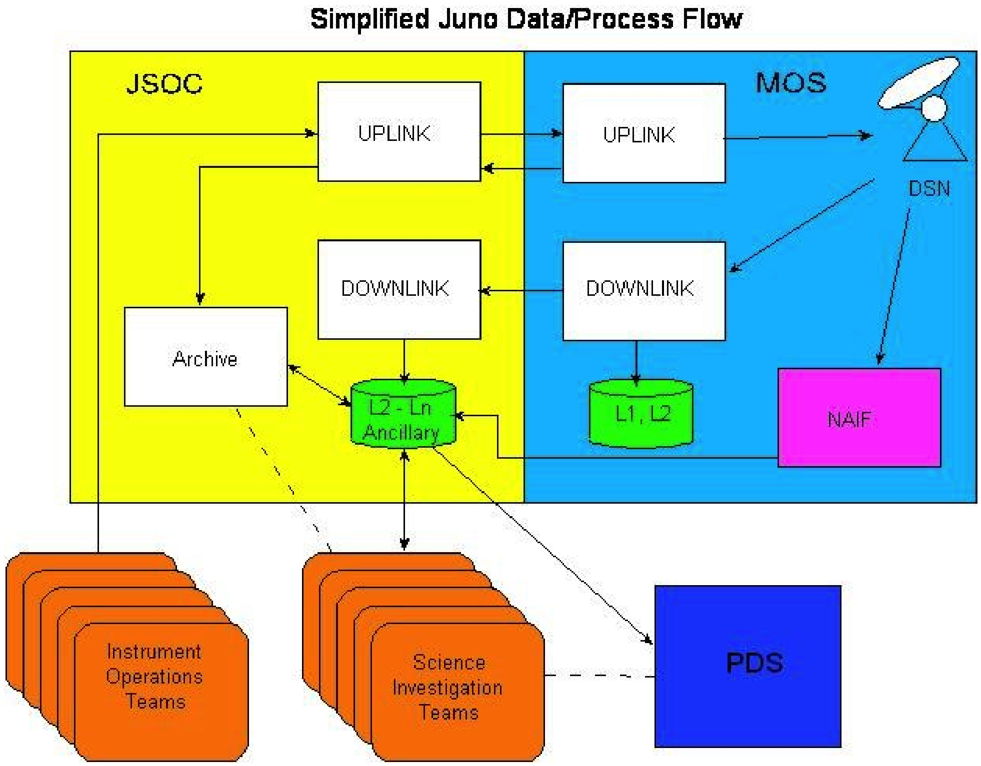


Figure 1: Juno science data flow diagram. White boxes are processes and solid arrows indicate data flow.

3.3 Data Processing and Production Pipeline

A single pipeline generates EDR records temporarily on route to generating RDR records, with the CODMAC level 2 data being the RDR records only, as shown in Figure 2.

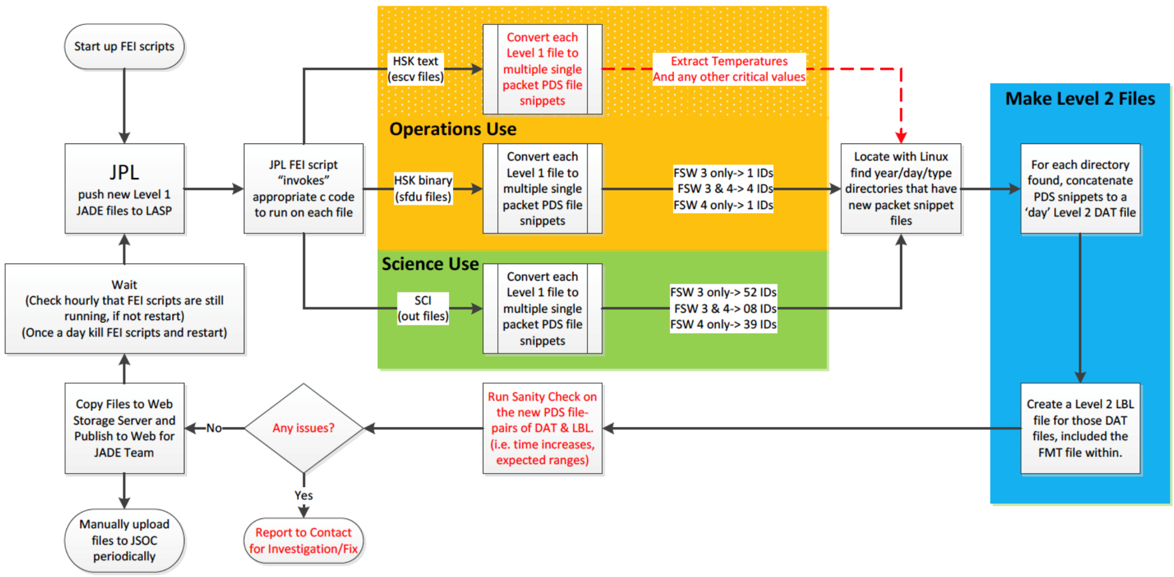


Figure 2: JADE science data pipeline diagram.

3.3.1 CODMAC Level 2 Data Production Pipeline

New data is pushed to LASP by FEI (JPL software) subscriptions to the CODMAC Level 1 data at JPL. The FEI subscriptions permanently run on a production computer hosted at LASP, with hourly cron checks to ensure the subscriptions are still running (and re-establish if necessary). In addition, a cron does a weekly check that all data available by FEI has been downloaded locally. Ingestion scripts can be triggered by FEI or by a regular cron job that looks for new local files. The binary files can contain any number of packets of any type of JADE data (PACKETIDs > 10), which are split out such that every unique packet is written to its own file – a PDS packet-snippet. The software that does that is written in c for speed, and every object in the packet is checked that it is within an allowed range expected for that particular packet. A few extra objects are added, such as a UTC object generated from the spacecraft clock values in the packet and converted with SPICE routines. PDS packet-snippets contain the time of day in the filename and are written in to directory structures based on date (*yyyy/yyyyddd/data_type/*), and are almost PDS compliant in that the DAT files obey the FMT file descriptions for the given product, but do not have a corresponding LBL file (LBL files are only generated once daily files are created). Duplicate packets in the FEI data originally over-wrote the previous PDS packet-snippet so only one is kept, but now a duplicate packet loads in the old one first, and if identical leaves it, otherwise reports an error as being different (a situation that has never occurred to date).

Reordering the data is now merely a cron to go through each *yyyy/yyyyddd/data_type/* directory, seeing if any files were modified/created since the last time the cron ran, and if so, concatenating the files of each product to a single file (with a Linux *cat* command). Due to the time of day being noted in the filenames, this concatenates them in the correct time order. The daily files are written to a different set of *yyyy/yyyyddd/data_type/* directories (see Figure 4), and code run to create the LBL file for each daily file. These are then fully PDS compliant CODMAC Level 2 daily files for each product available that day, ready for upload to JSOC via FTP.

3.3.2 CODMAC Level 3 Data Production Pipeline

The Level 2 data (previous section) are used as input to generate the Level 3 files, together with reconstructed SPICE kernels in order to provide position and orientation information (see section 6.2.10.5 for more details about the conversion to science units). This is done in IDL, and can take several minutes to run per file, as such a cron job will run nightly (if not more often) to call IDL to create level 3 DAT files for any new level 2 files that have appeared in the last day. The same cron job then runs a python code that generates the corresponding LBL files (containing the FMT file) for the DAT file and pushes them to the appropriate *yyyy/yyyyddd/data_type/* directories (see Figure 4). These are then fully PDS compliant CODMAC Level 3 daily files for each product available that day, ready for upload to JSOC via FTP.

3.3.3 CODMAC Level 4 Data Production Pipeline

JADE has no Level 4 products (see Table 5), so jumps straight from Level 3 to Level 5 products.

3.3.4 CODMAC Level 5 Data Production Pipeline

The Level 3 data (previous section) are used as input to generate the Level 5 binary files, together with any required reconstructed SPICE kernels (if that data was not already in the level 3 file) in order to provide position and orientation information (see section 6.2.10.5 for more details about the conversion to science units). Level 3 Juno MAG files from the PDS are also a required input. This is done in IDL, and can take several minutes to run per file. Since this required MAG files from the PDS, it is run only when new MAG files appear on the PDS, usually at 2-orbit intervals. Those runs call IDL to create level 5 DAT files for any existing JADE level 3 files that have yet to be converted to level 5. A python code then generates the corresponding LBL files (containing the FMT file) for the DAT file and pushes them to the appropriate *yyyy/yyyyddd/data_type/* directories (see Figure 5). These are then fully PDS compliant CODMAC Level 5 daily binary files for each product available that day, ready for upload to JSOC via FTP.

The Level 5 ASCII files use the Level 5 binary files as input, and thus are made on the same schedule as the Level 5 binary files. IDL codes are run to make the comma separated variable (CSV) output files, which depending on the input file (and if the source was high rate or low rate data) can take from a few minutes per file, to nearly an hour, to run. A python code then generates the corresponding LBL files (containing the FMT file) for the CSV file and pushes them to the appropriate *yyyy/yyyyddd/* directories (see Figure 5). These are then fully PDS compliant CODMAC Level 5 daily ASCII files for each product available that day, ready for upload to JSOC via FTP.

3.4 Data Validation

General PDS syntax / formatting checks are carried out at the earliest stage of processing to ensure the data obeys the PDS standards. Periodically PDS online validation tools are run on local volumes to ensure that the CODMAC Level 2/3/5 product standards are met, as well as bespoke checks carried out locally. This ensures values are within acceptable ranges (or a fill value, a.k.a. MISSING_CONSTANT), correctly ordered, and correctly labeled.

3.4.1 Instrument Team Validation

The JADE instrument team will be the first to see any data by monitoring trend plots and examining the data to ensure what was commanded occurred. In addition they will monitor the health of the various sensors and carry out regular calibration exercises. During these activities any inconsistencies that may arise will be investigated, corrected where possible, or noted in the `ERRATA.TXT` for the volume.

3.4.2 Science Team Validation

The JADE science team will provide validation by virtue of using the data and reporting any inconsistencies to the instrument team. Since each orbit takes up to 53 days and there is a 3 to 6 month lag between acquiring the data and providing it to the PDS (see Table 8) there is enough time for the science team to work with the data prior to PDS submission.

4 Archive volume generation

The JADE Standard Data Record archive collection is produced by the JADE IOT in cooperation with the JSOC, and with the support of the PDS Planetary Plasma Interactions (PPI) Node at the University of California, Los Angeles (UCLA). The archive volume creation process described in this section sets out the roles and responsibilities of each of these groups. The assignment of tasks has been agreed by all parties. Archived data received by the PPI Node from the JADE team will be made electronically available to PDS users as soon as practical but no later than as laid out in Table 8.

4.1 Data transfer methods and delivery schedule

The JADE team will deliver data to the PPI Node in standard product packages containing three months of data, also adhering to the schedule set out in Table 8. Each package will comprise both data and ancillary data files organized into directory structures consistent with the volume design described in Section 5, and combined into a deliverable file(s) using file archive and compression software. When these files are unpacked at the PPI Node in the appropriate location, the constituent files will be organized into the archive volume structure.

Table 8: Archive Schedule and Responsibilities

Instrument	Data Product	Provider	Earth Flyby (EFB)	Other Cruise	Orbital Phase
JADE R.J. Wilson	P0 (Level 2)	JADE Team	*JADE was off throughout EFB, => No data to archive.*	Jupiter + 4 mo.	EDA + 3 to 6 mo.
	P1, P2 (Level 3)	JADE Team		Jupiter + 4 mo.	EDA + 3 to 6 mo.
	P3, P4, P5, P6, P7 (Level 5)	JADE Team		Jupiter + 4 mo.	Deorbit + 9 mo.

EDA – End of data acquisition

The archives will be sent electronically from the JADE IOT to a user account on the PPI node using the *ssh* protocol. The IOT operator will copy each volume (see Table 10) in the form of a compressed *tar* archive (a.k.a. *tarball*) to an appropriate location within the PPI file system. Only those files that have changed since the last delivery will be included. The PPI operator will decompress the data, using the *tar* checksums to verify that the archive is complete. Each step of data submission process will be tracked in a version CATS (Cassini Archive Tracking System) which has been adapted for use by Juno.

Following receipt of a data delivery, PPI will organize the data into PDS archive volume structure within its online data system. PPI will generate all of the required files associated with a PDS archive volume (index file, read-me files, etc.) as part of its routine processing of incoming JADE data. Newly delivered data will be made available publicly through the PPI online system once accompanying labels and other documentation have been validated. It is anticipated that this validation process will require no more than fourteen working days from receipt of the data by PPI. The first two data deliveries are expected to require somewhat more time for the PPI Node to process before making the data publicly available.

The Juno prime mission begins after JOI and two subsequent correction orbits, and lasts for 33 ~53 day orbits. Table 8 formalizes the data delivery schedule for the entire Juno mission, including cruise, commissioning and prime mission phases. Data delivery from JSOC to PPI node will occur on the 15th of the month and the data will be publicly available on the 1st of the following month. Archiving of products from any extended mission period will be negotiated with the Project at a later date. [Juno began an extended mission in 2021, but at time of writing, specific delivery dates are still under negotiation.]

4.2 Data validation

The JADE standard data archive volume set will include all data acquired during the Juno mission. The archive validation procedure described in this section applies to volumes generated during both the cruise and prime phases of the mission.

PPI node staff will carefully examine the first archive volume that they receive that contains data from JADE to determine whether the archive is appropriate to meet the stated science objectives of the instrument. The PPI node will also review the archive product generation process for robustness and ability to detect discrepancies in the end products; documentation will be reviewed for quality and completeness.

As expertise with the instrument and data develops the JADE team may decide that changes to the structure or content of its standard data products are warranted. Should these changes be implemented, the new data product and archive volume will be subjected to a full PDS peer review, and this document will be revised to reflect the modified archive. Table 2 lists the history of all modifications to the archive structure and contents.

4.3 Data product and archive volume size estimates

JADE standard data products are organized into files that span 24 hours, breaking at 0h UTC. Files vary in size depending on the telemetry rate and allocation. Table 9 summarizes the expected sizes of the JADE standard products (as estimated in 2017, see table footnoted d and e) and are per version number of data. The extended mission orbits are shorter than 53 days so should have less data on average, although may have a similar volume of high-rate science data per orbit. Table 9 includes actual numbers as of late 2022 (where possible) rather than estimates. The total size for prime mission is not as simple as the per orbit multiplied by number of orbits since the number of high rate science (HRS) hours per orbit, and the low rate science accumulation times used per orbit varied greatly (see footnote d).

Table 9 is an estimate for the primary mission, we would expect the volume production rate for the extended mission phase to be similar again (more orbits, but shorter durations per orbit).

All JADE standard data are organized by the PDS team onto a single archive volume. The data on the volume are organized into one-day subdirectories.

Table 9: Data product size and archive volume production rate (per Level (L) and version (V))

Data Product	Production rate per day[b] (approx.)	Production rate per orbit[a] (approx.) (assumes 53 day orbits)	'Primary' mission size 2015 to PJ34 (2021-159)
L2 V01 Science	265 MB	13.7 GB[d]	198 GB
L3 V01, V02, V03 Science	2878 MB	149.0 GB[d]	2336 GB[e]
L3 V04[e] Science	1634 MB	84.6 GB[d]	1226 GB
L5 Science (Binary)	1543 MB	80 GB[d]	1136 GB
L5 Science (ASCII)[f]	0.32 MB	17 MB[d]	202 MB
Total (L2+L3V04+L5)	**3443 MB[b]**	**179 GB[a]**	**2560 GB**

MB = Megabyte, GB = Gigabyte

a. *Per orbit value based on full day data from PJ4 to PJ5-1 day (2017 DOY 033-085 inclusive), which included 6 hours of HRS.*
b. *Per day value is per orbit divided by 53 for this table, despite later, EM, orbits being shorter.*
c. *Typically 6 hours[d] of HRS data per 53-day orbit, usually on the same day, hence the daily production rate (as an average or per orbit) is greater than the median daily production rate.*
d. *[Update in 2021] In later orbits there was much more HRS data opportunity than the 6 hours we originally hoped for, hence values in this table (top half, from early prime mission) are significantly under estimated. Later Prime mission orbits often had about 17 hours of HRS on average, but could range over 12 to 31.5 hours for specific orbits.*
e. *Level 3 (L3) Version 04 data was first created in late 2021 in time for the PJ34 delivery to complete the prime mission phase. Versions 01 to 03 all stopped at earlier dates (but after PJ5) so do not have a full prime mission dataset to measure.*
f. *L5 ASCII files do not cover all time intervals, but hand-picked 'good intervals' of each day, if any on a given day.*

Following receipt of JADE data by the PPI Node it is expected that fourteen working days will be required to validate and process the delivery before the data are made available on PPI web pages. New deliveries will be added to the existing volume structure to which they belong.

4.4 Backups and duplicates

The PPI Node keeps three copies of each archive volume. One copy is the primary online archive, another is an onsite backup copy, and the final copy is an off-site backup copy. Once the archive volumes are fully validated and approved for inclusion in the archive, a copy of the data is sent to the National Space Science Data Center (NSSDC) for long-term archive in a NASA-approved deep-storage facility. The PPI Node may maintain additional copies of the archive volumes, either on or off-site as deemed necessary. The process for the dissemination, and preservation JADE archive volumes is illustrated in Figure 3

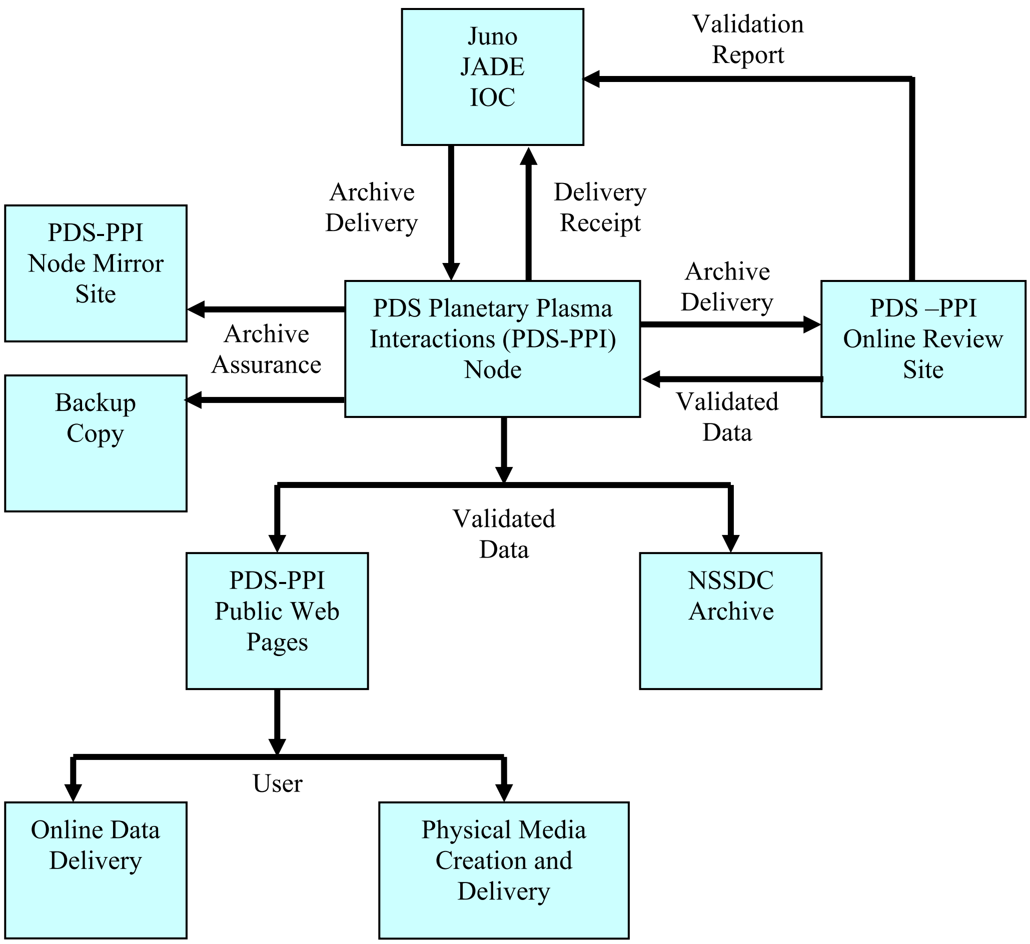


Figure 3: Duplication and dissemination of JADE standard archive volumes.

4.5 Labeling and identification

Each JADE data volume bears a unique volume ID using the last two components of the volume set ID [*PDS Standards Reference*, see §19]. For each physical medium, the volume IDs are USA_NASA_PDS_??????_*mnnn*, where ?????? is the VOLUME_SET_ID defined by the PDS and *mnnn* is the sequence number of the individual volume, where the *m* refers to the CODMAC level of the data. Hence the first JADE Level 2 volume has the volume ID JNOJAD_2002, as shown in Table 10. (JNOJAD_2001 was used for an earlier FSW version covering 2011-2014, far before Jupiter and all for calibration/engineering use, see note in table.)

Table 10: PDS Data Set Volume Assignments

Level	DATA_SET_ID	VOLUME_ID
2	JNO-SW-JAD-2-UNCALIBRATED-V1.0 JNO-J/SW-JAD-2-UNCALIBRATED-V1.0	JNOJAD_2001 JNOJAD_2002
3	JNO-J/SW-JAD-3-CALIBRATED-V1.0	JNOJAD_3001
5	JNO-J/SW-JAD-5-CALIBRATED-V1.0	JNOJAD_5001
5	JNO-J-JAD-5-MOMENTS-V1.0	JNOJAD_5002

Note: JNOJAD_2001 is described in a separate SIS for FSW3 (2011-2014), found in that volume.

5 Archive volume contents

This section describes the contents of the JADE standard product archive collection volumes, including the file names, file contents, file types, and the organizations responsible for providing the files. The complete directory structure is shown in Figure 4 (for level 2 and 3 volumes) and Figure 5 (for level 5 volumes). All the ancillary files described herein appear on each JADE standard product volume, except where noted (highlighted yellow).

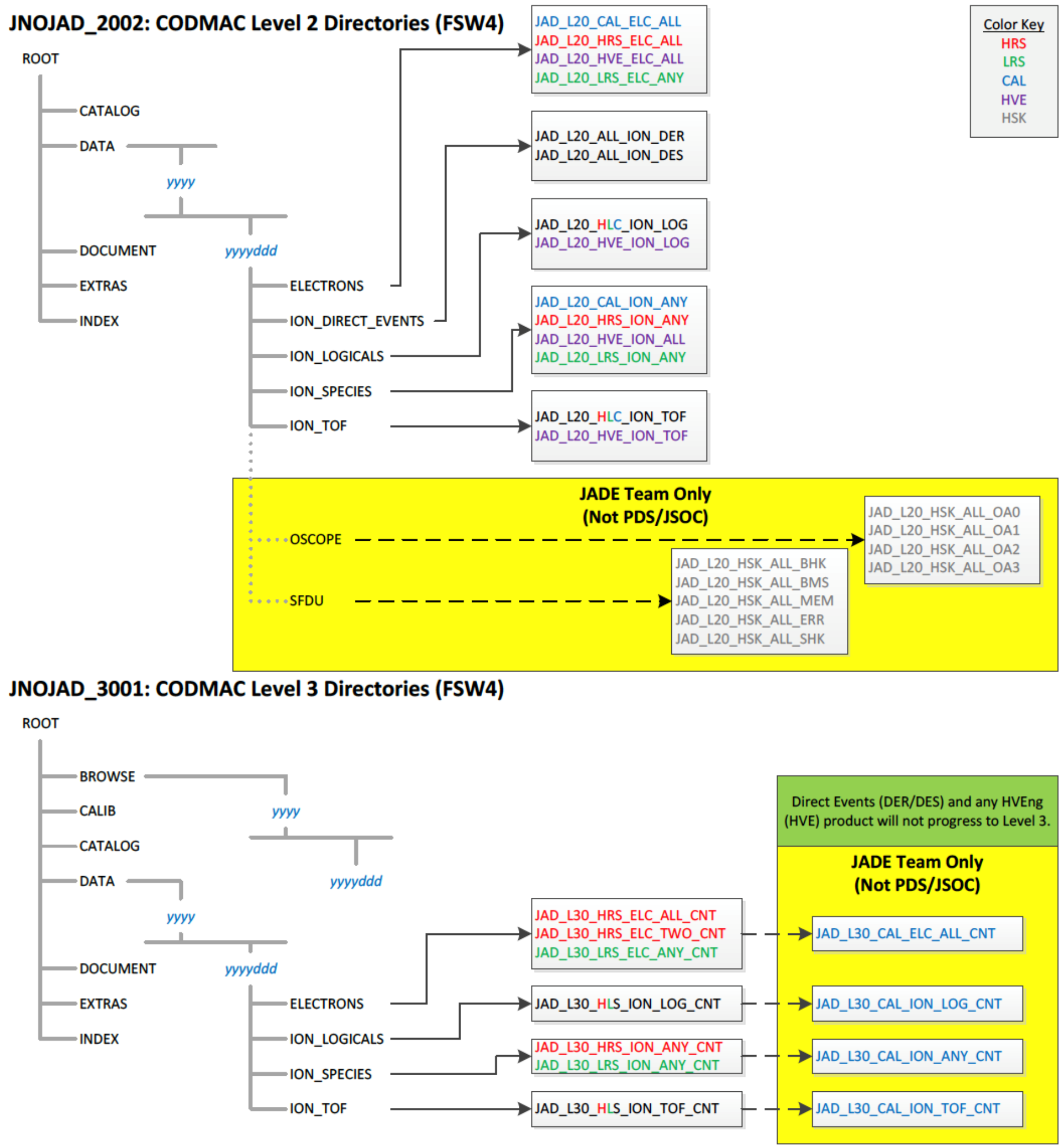
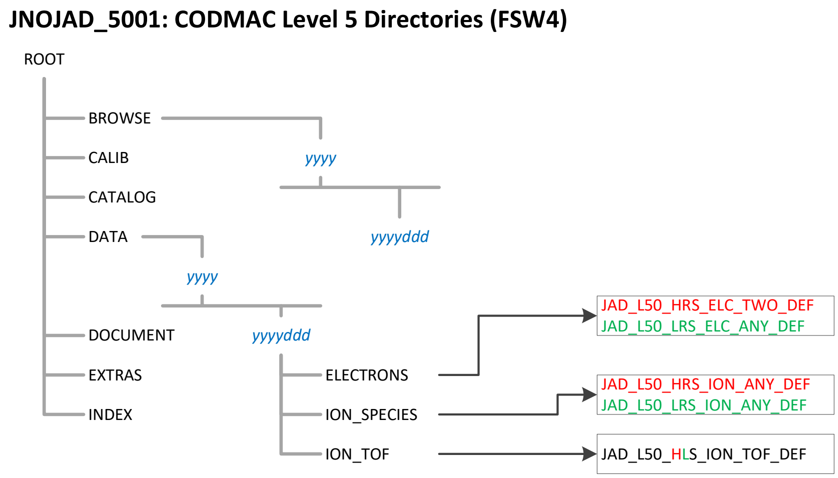


Figure 4: Archive volume directory structure for Levels 2 and 3
(File names shown, not STANDARD_DATA_PRODUCT_ID.)



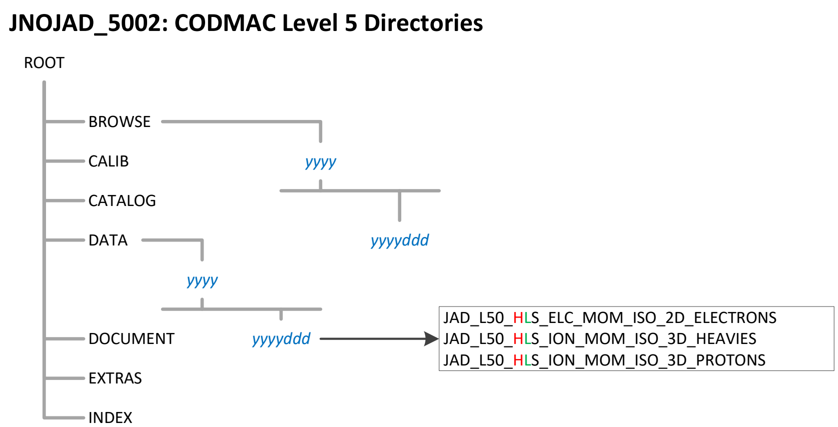


Figure 5: Archive volume directory structure for Level 5 datasets
(File names shown, not STANDARD_DATA_PRODUCT_ID.)
(Color key same as that in Figure 4.*)*

5.1 Root directory

The files listed in Table 11 are contained in the (top-level) root directory, and are produced by the JADE team in consultation with the PPI node of the PDS. With the exception of the hypertext file and its label, all of these files are required by the PDS volume organization standards.

Table 11: Root directory contents

File	Description	Responsibility
AAREADME.TXT	This file completely describes the volume organization and contents (PDS label attached)	PPI
ERRATA.TXT	A text file containing a cumulative listing of comments and updates concerning all JADE standard products on all JADE volumes in the volume set published to date	JADE team
VOLDESC.CAT	A description of the contents of this volume in a PDS format readable by both humans and computers	PPI

5.2 BROWSE directory (Not for Level 2 Volume)

The BROWSE directory contains [TBD granularity] browse plots of the JADE data, split into [TBD] intervals. The contents of this directory and its subdirectories are described in Table 12.

Table 12: BROWSE directory contents

File	Description	Responsibility
BROWINFO.TXT	A description of the contents of this directory	PPI
[TBD]	[TBD]	

5.3 CALIB directory (Level 3 Volume Only)

The CALIB directory, which only exists on the CODMAC Levels 3 archives, contains a copy of the calibration plan and the ancillary data used to calibrate the JADE instrument performance. The contents of this directory are described in Table 13 (over two pages), where every file has a corresponding label (*.LBL) file that is not listed in the table. Most files are comma separated variable (csv) files so that they are easy to open in a text editor or spreadsheet, and are based on the look up table (LUT) version uploaded to the spacecraft for a given time.

Table 13: CALIB directory contents

File	Description	Respon-sibility
CALINFO.TXT	A description of the contents of this directory.	PPI
ANODE_LOOK_ELC_DEFL_EQNS_V*vv*.PDF	PDF of Electron sensor anode elevation look direction equations for when deflectors are on (HRS at Jupiter only). Version number *vv*. Azimuths are unaffected, in file: ANODE_LOOK_ELC_DEFL_NONE_V*vv*.CSV	JADE team
ANODE_LOOK_ELC_DEFL_NONE_V*vv*.CSV	Electron sensors anodes look directions in spacecraft azimuth and elevation when the deflectors are off. Version number *vv*. Lower, center and upper values (in degrees).	JADE team
ANODE_LOOK_ION_DEFL_NONE_V*vv*.CSV	Ion sensor anodes look directions in spacecraft azimuth and elevation when the deflectors are off. Version number *vv*. Lower, center and upper values (in degrees).	JADE team
DATA_UNCERTAINTY_EQNS_V*vv*.PDF	PDF of the method and equations used to generate the DATA_SIGMA values in the level 3 files. Version number *vv*.	JADE team
JADE_LEVEL3_V*mm*_COMPARED_TO_V*nn*_DESCRIPTION_V*vv*.PDF	Describes the differences between Level 3 version *mm* and Level 3 version *nn* files. Document version number *vv*. Note: *mm* = *nn* + 1.	JADE team
JAD_L30_CALIB_LIST_*nnnnn*.TXT	List of JADE calibration files or methods used to generate level 3 products, version *nnnnn*. (May be a PDF version if file is not present.)	JADE team
JAD_L30_CALIB_LIST_*nnnnn*.PDF	List of JADE calibration files or methods used to generate level 3 or level 5 products, version *nnnnn*. [PDF allows figures and equations that would be tricky in above .TXT file.]	JADE team
JAD_L30_SPICE_METAKERNEL_*nnnnn*.TXT	SPICE metakernel used to generate level 3 products, version *nnnnn*. (The individual SPICE kernels can be downloaded from NAIF.)	JADE team

File	Description	Respon-sibility
LUT_*m*_*nn*_COMPRESSION.CSV e.g. LUT_*3*_*00*_COMPRESSION.CSV	LUT *m*.*nn* (e.g. LUT *3*.*00*), 16->8 bit and 32->8 bit compression tables. (No version number as tables uploaded to s/c.)	JADE team
LUT_*m*_*nn*_ENERGY_V*vv*.CSV e.g. LUT_*3*_*00*_ENERGY_V*01*.CSV	LUT *m*.*nn* (e.g. LUT *3*.*00*), Version number *vv*. Ion and electron sensors E and ΔE/E tables. (Note, LUT 3.08 is never used in flight.) Some LUTs have energy tables that vary with time, see LUT_*m*_*nn*_*T*_ENERGY_V*vv* files.	JADE team
LUT_*m*_*nn*_*T*_ENERGY_V*vv*.CSV e.g. LUT_*4*_*01*_*A*_ENERGY_V*01*.CSV	LUT *m*.*nn* (e.g. LUT *3*.*00*), Time Period *T* (= A,B,C,…), Version number *vv*. Ion and electron sensors E and ΔE/E tables. LBL files contain the start/stop time of each period. If there is no time dependence during a LUT, use LUT_*m*_*nn*_ENERGY_V*vv* files.	JADE team
LUT_*m*_*nn*_TOF_SPECIES_MAP.CSV e.g. LUT_*3*_*00*_TOF_SPECIES_MAP.CSV	LUT *m*.*nn* (e.g. LUT *3*.*00*), which TOF channels map to ion species 3, 4 & 5 for a given energy step. (No version number as tables uploaded to s/c.)	JADE team
TOF_CHANNEL_TO_SECONDS_HLC_V*vv*.CSV	Convert the 96 TOF channels numbers to seconds. Version number *vv*. (The Level 3 TOF data files contain these values.)	JADE team
TOF_CHANNEL_TO_SECONDS_HVE_V*vv*.CSV	Convert the 128 TOF channels numbers to seconds. Version number *vv*. (For Level 2 HVE TOF.)	JADE team

5.4 CATALOG directory

The files in the CATALOG directory provide a top-level understanding of the Juno mission, spacecraft, instruments, and data sets in the form of completed PDS templates. The information necessary to create the files is provided by the JADE team and formatted into standard template formats by the PPI Node. The files in this directory are coordinated with PDS data engineers at both the PPI Node and the PDS Engineering Node.

Table 14: CATALOG directory contents

File	Description	Responsibility
CATINFO.TXT	A description of the contents of this directory	PPI
JADE_INST.CAT	PDS instrument catalog description of the JADE instrument	JADE team
JADE_DS.CAT	PDS data set catalog description of the JADE data files	Initial: JADE team Up-keep: PPI
JADE_REF.CAT	JADE-related references mentioned in other CAT files	JADE team
INSTHOST.CAT	A description of the Juno spacecraft	Juno Project
MISSION.CAT	PDS mission catalog description of the Juno mission	Juno Project
PERSON.CAT	PDS personnel catalog description of JADE team members and other persons involved with generation of JADE standard data products	JADE team
PROJ_REF.CAT	References mentioned in INSTHOST.CAT and MISSION.CAT	Juno Project

5.5 DATA directory

5.5.1 Contents

The DATA directory contains the data files produced by the JADE team. In the Level 2 archive, these files contain the raw binary instrument EDR's, organized into correct time sequence, time tagged, and edited to remove obviously bad data. In the higher level archives, the contents of the DATA directory are binary files that result from passing the corresponding Level 2 files through the processing pipeline.

The data files are of the highest quality possible. Any residual issues are documented in AAREADME.TXT and ERRATA.TXT files in the volume's root directory, or in JADE_DS.CAT in the CATALOG directory. Users are referred to these files for a detailed description of any outstanding matters associated with the archived data.

Table 15: DATA directory contents

File	Description	Responsibility
yyyy	Subdirectories containing JADE data acquired in year *yyyy*	JADE team

5.5.2 Subdirectory structure

In order to manage files in an archive volume more efficiently the DATA directory is divided into subdirectories. The two levels of division are based on time; data are organized into yearly subdirectories, which are further divided into a number of daily sub-subdirectories. The naming convention for the yearly directories is *yyyy*, and for the daily directories it is *yyyyddd*, where *ddd* is the three-digit day of year. For example, all data for the year 2015 are contained below the directory 2015, with data for Jan 1 2015 UTC found in the subdirectory 2015/2015001, and so on.

'Daily' files are from UTC midnight to midnight, where the day of year at the start of each JADE record defines which day of year that record is assigned to; e.g. if a JADE record had a 30 second accumulation time starting at 2015-015T23:59:54, then the record is considered to be part of 2015-015, despite the center time being 2015-016T00:00:09, or end time being 2015-016T00:00:24. Level 2 records UTC times are always start times. Level 3 records provide start (lower), center and end (upper) UTC times separately.

5.5.3 Required files

A PDS label describes each file in the DATA path of an archive volume. Text documentation files have attached (internal) PDS labels and data files have detached labels. Detached PDS label files have the same root name as the file they describe but have the extension LBL. The label files contain both data file content and record structure information.

5.5.4 The *yyyy/yyyyddd* subdirectory

This directory contains JADE data files and their corresponding PDS labels. As shown in Table 16, Table 17, Table 18 and Table 19 for CODMAC levels 2, 3, 5 (binary files) and 5 (ASCII files) respectively, the data in these files span a time interval of one day, the particular day being identified from both the file name and the name of the parent directory. The names also contain a 2-digit version. The initial version is V01.

Binary data file names have the "DAT" file extension. ASCII data file names have the "CSV" file extension. Each file is accompanied by a PDS label (LBL) describing its contents. The labels permit the contents of most of the products to be browsed by PDS software, e.g., *NASAView*, etc.

Table 16: CODMAC Level 2 DATA/yyyy/yyyyddd directory contents

Filename	Description
ELECTRONS	Subdirectories containing JADE electron data (all electron sensors) acquired for year/doy *yyyyddd*.
ION_DIRECT_EVENTS	Subdirectories containing JADE ion Direct Event data acquired for year/doy *yyyyddd*.
ION_LOGICALS	Subdirectories containing JADE ion Logicals data acquired for year/doy *yyyyddd*.
ION_SPECIES	Subdirectories containing JADE ion species data (for various ion species) acquired for year/doy *yyyyddd*.
ION_TOF	Subdirectories containing JADE ion time of flight data acquired for year/doy *yyyyddd*.

Table 17: CODMAC Level 3 DATA/yyyy/yyyyddd directory contents

Filename	Description
ELECTRONS	Subdirectories containing JADE electron data (all electron sensors) acquired for year/doy *yyyyddd*.
ION_LOGICALS	Subdirectories containing JADE ion Logicals data acquired for year/doy *yyyyddd*.
ION_SPECIES	Subdirectories containing JADE ion species data (for various ion species) acquired for year/doy *yyyyddd*.
ION_TOF	Subdirectories containing JADE ion time of flight data acquired for year/doy *yyyyddd*.

Table 18: CODMAC Level 5 DATA/yyyy/yyyyddd directory contents (with Binary files)

Filename	Description
ELECTRONS	Subdirectories containing JADE electron data (all electron sensors) acquired for year/doy *yyyyddd*.
ION_SPECIES	Subdirectories containing JADE ion species data (for various ion species) acquired for year/doy *yyyyddd*.
ION_TOF	Subdirectories containing JADE ion time of flight data acquired for year/doy *yyyyddd*.

Table 19: CODMAC Level 5 DATA/yyyy/yyyyddd directory (with ASCII files)

Filename	Description
JAD_L50_HLS_ELC_MOM_ISO_2D_ELECTRONS_*yyyyddd*_V*nn*.CSV	Electron Moments, 2 dimensional with isotropic pressure and temperature.
JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES_*yyyyddd*_V*nn*.CSV	Heavy Ion Moments, 3 dimensional with isotropic pressure and temperature.
JAD_L50_HLS_ION_MOM_ISO_3D_PROTONS_*yyyyddd*_V*nn*.CSV	Proton Moments, 3 dimensional with isotropic pressure and temperature.

5.5.4.1 The *ELECTRONS* subdirectory

This directory contains JADE data files from the electron sensors and their corresponding PDS labels. As shown in Table 20, Table 21 and Table 22, the data in these files span a time interval of one day, the particular day being identified from both the file name and the name of the parent directory. The names also contain a 2-digit version. The initial version is V01. This directory exists in CODMAC level 2, 3 and 5 (binary files) volumes.

Table 20: DATA/yyyy/yyyyddd/ELECTRONS directory contents for Level 2 data

Filename	Description
JAD_L20_CAL_ELC_ALL_*yyyyddd*_V*nn*.DAT	MCP calibration mode electron counts, all 3 sensor.
JAD_L20_HRS_ELC_ALL_*yyyyddd*_V*nn*.DAT	High Rate Science electron counts, all 3 sensors.
JAD_L20_HVE_ELC_ALL_*yyyyddd*_V*nn*.DAT	High Voltage Engineering electron counts, all 3 sensors.
JAD_L20_LRS_ELC_ANY_*yyyyddd*_V*nn*.DAT	Low Rate Science electron count rate, any of the 3 electron sensor.

Table 21: DATA/yyyy/yyyyddd/ELECTRONS directory contents for Level 3 data

Filename	Description
JAD_L30_HRS_ELC_ALL_CNT_*yyyyddd*_V*nn*.DAT	High Rate Science electron counts per second, all 3 sensors per record.
JAD_L30_HRS_ELC_TWO_CNT_*yyyyddd*_V*nn*.DAT	High Rate Science electron counts per second, for E060 and E180, not E300.
JAD_L30_LRS_ELC_ANY_CNT_*yyyyddd*_V*nn*.DAT	Low Rate Science electron counts per second, just one of the 3 sensors per record (see record for which one).

JADE-E300 was turned off in early 2016. The Level 3 JAD_HRS_ELC_ALL_CNT_* files exist for all HRS data in file versions *01*, *02* and *03*. However, from Level 3 Version *04*, these JAD_HRS_ELC_ALL_CNT_* files will only exist for days when JADE-E300 was on.

Table 22: DATA/yyyy/yyyyddd/ELECTRONS directory contents for Level 5 binary data

Filename	Description
JAD_L50_HRS_ELC_TWO_DEF_*yyyyddd*_V*nn*.DAT	High Rate Science electron Differential Energy Flux, for E060 and E180, not E300.
JAD_L50_LRS_ELC_ANY_DEF_*yyyyddd*_V*nn*.DAT	Low Rate Science electron Differential Energy Flux, just one of the 3 sensors per record (see record for which one).

5.5.4.2 The *ION_DIRECT_EVENTS* subdirectory

This directory contains JADE data files from ion direct events and their corresponding PDS labels. As shown in Table 23, the data in these files span a time interval of one day, the particular day being identified from both the file name and the name of the parent directory. The names also contain a 2-digit version. The initial version is V01. This is a CODMAC level 2 only directory.

Table 23: DATA/yyyy/yyyyddd/ION_DIRECT_EVENTS directory contents

Filename	Description
JAD_L20_ALL_ION_DER_*yyyyddd*_V*nn*.DAT	Direct events (raw), for high and low rate science, calibration and high voltage engineering modes.
JAD_L20_ALL_ION_DES_*yyyyddd*_V*nn*.DAT	Direct events (split out), for high and low rate science, calibration and high voltage engineering modes.

5.5.4.3 The *ION_LOGICALS* subdirectory

This directory contains JADE data files from ion Logicals and their corresponding PDS labels. As shown in Table 24 and Table 25, the data in these files span a time interval of one day, the particular day being identified from both the file name and the name of the parent directory. The names also contain a 2-digit version. The initial version is V01. This directory exists in both CODMAC level 2 and 3 volumes.

Table 24: DATA/yyyy/yyyyddd/ION_LOGICALS directory contents for Level 2 data

Filename	Description
JAD_L20_HLC_ION_LOG_*yyyyddd*_V*nn*.DAT	Ion Logical counts, for high and low rate science plus calibration modes.
JAD_L20_HVE_ION_LOG_*yyyyddd*_V*nn*.DAT	Ion Logical counts, for the high voltage-engineering mode.

Table 25: DATA/yyyy/yyyyddd/ION_LOGICALS directory contents for Level 3 data

Filename	Description
JAD_L30_HLS_ION_LOG_CNT_*yyyyddd*_V*nn*.DAT	Ion Logical counts per second, for high and low rate modes.

5.5.4.4 The *ION_SPECIES* subdirectory

This directory contains JADE data files and their corresponding PDS labels. As shown in Table 26, Table 27 and Table 28, the data in these files span a time interval of one day, the particular day being identified from both the file name and the name of the parent directory. The names also contain a 2-digit version. The initial version is V01. This directory exists in CODMAC level 2, 3 and 5 (binary files) volumes.

Table 26: DATA/yyyy/yyyyddd/ION_SPECIES directory contents for Level 2 data

Filename	Description
JAD_L20_CAL_ION_ANY_*yyyyddd*_V*nn*.DAT	Calibration mode ion species count rate.
JAD_L20_HRS_ION_ANY_*yyyyddd*_V*nn*.DAT	High rate science ion species counts.
JAD_L20_HVE_ION_ALL_*yyyyddd*_V*nn*.DAT	High Voltage Engineering for all ion species counts.
JAD_L20_LRS_ION_ANY_*yyyyddd*_V*nn*.DAT	Low Rate Science ion species count rate.

Table 27: DATA/yyyy/yyyyddd/ION_SPECIES directory contents for Level 3 data

Filename	Description
JAD_L30_HRS_ION_ANY_CNT_*yyyyddd*_V*nn*.DAT	High Rate Science ion species counts per second.
JAD_L30_LRS_ION_ANY_CNT_*yyyyddd*_V*nn*.DAT	Low Rate Science ion species count per second.

Table 28: DATA/yyyy/yyyyddd/ION_SPECIES directory contents for Level 5 binary data

Filename	Description
JAD_L50_HRS_ION_ANY_DEF_*yyyyddd*_V*nn*.DAT	High Rate Science ion species Differential Energy Flux.
JAD_L50_LRS_ION_ANY_DEF_*yyyyddd*_V*nn*.DAT	Low Rate Science ion species Differential Energy Flux.

5.5.4.5 The *ION_TOF* subdirectory

This directory contains JADE data files and their corresponding PDS labels. As shown in Table 29, Table 30 and Table 31, the data in these files span a time interval of one day, the particular day being identified from both the file name and the name of the parent directory. The names also contain a 2-digit version. The initial version is V01. This directory exists in CODMAC level 2, 3 and 5 (binary files) volumes.

Table 29: DATA/yyyy/yyyyddd/ION_TOF directory contents for Level 2 data

Filename	Description
JAD_L20_HLC_ION_TOF_*yyyyddd*_V*nn*.DAT	Ion time of flight counts, for high and low rate science plus calibration modes.
JAD_L20_HVE_ION_TOF_*yyyyddd*_V*nn*.DAT	Ion time of flight counts, for the high voltage-engineering mode.

Table 30: DATA/yyyy/yyyyddd/ION_TOF directory contents for Level 3 data

Filename	Description
JAD_L30_HLS_ION_TOF_CNT_*yyyyddd*_V*nn*.DAT	Ion time of flight counts per second, for high and low rate modes.

Table 31: DATA/yyyy/yyyyddd/ION_TOF directory contents for Level 5 binary data

Filename	Description
JAD_L50_HLS_ION_TOF_DEF_*yyyyddd*_V*nn*.DAT	Ion time of flight Differential Energy Flux, for high and low rate modes.

5.6 DOCUMENT directory

The DOCUMENT directory contains a range of documentation considered either necessary or useful for users to understand the archive data set. Documents may be included in multiple forms, for example, ASCII, PDF, or HTML. PDS standards require that any documentation needed for use of the data be available in an ASCII format. "Clean" HTML is an acceptable ASCII format in addition to plain text. "Clean" HTML refers to HTML with minimal markup, and formatted in such a way as to facilitate reading in a text browser. Table 32 describes the contents of the DOCUMENT directory.

Table 32: DOCUMENT directory contents

Filename	Description	Responsibility
DOCINFO.TXT	A description of the contents of this directory	PPI
JADE_INST_PAPER.LBL	A PDS detached label for the JADE Instrument paper	JADE team
JADE_INST_PAPER.PDF	PDF version of the published paper (open access)	JADE team
JADE_INST_PAPER.HTM	The JADE instrument paper in HTML format (HTML Version 3.2)	JADE team
JADE_INST_PAPER_IMAGE_ *mmm*.JPG/PNG	Image files for JADE_INST_PAPER.HTM, some are JPG, others PNG, where *mmm* is a non-repeating incrementing number from 001 to 116.	JADE team
JADE_INST_PAPER_TABLE_ *mm*.PNG	Image files of the 18 tables from JADE_INST_PAPER.PDF, where *mm* is a non-repeating incrementing number from 01 to 18.	JADE team
JADE_FSW4_SIS_V*mm*.LBL	A PDS detached label for the SIS document, version *mm*.	JADE team
JADE_FSW4_SIS_V*mm*.DOCX	The SIS in Microsoft Word format; this was used to make the PDF – which is the file of record.	JADE team
JADE_FSW4_SIS_V01.HTM	The SIS in a simple HTML format. (Saved from Word, then hand edited. Only done for V01, future versions will use the PDF as the primary document of record, so no PNG files either.)	JADE team
JADE_FSW4_SIS_EQN_ *nn*.PNG (Only for JADE_FSW4_SIS_V01.HTM)	Image files of 6 equations (for the HTML file) from JADE_FSW4_SIS_V01.PDF, where *nn* is a non-repeating incrementing number from 01 to 06. (Note the SIS does not number equations.)	JADE team
JADE_FSW4_SIS_FIG_ *nn*.PNG (Only for JADE_FSW4_SIS_V01.HTM)	Image files of the 11 figures from JADE_FSW4_SIS_V01.PDF, where *nn* is a non-repeating incrementing number from 01 to 11. (Note that *nn* may not map to SIS figure nn.)	JADE team
JADE_FSW4_SIS_V*mm*.PDF	SIS in PDF format, the SIS version of record.	JADE team
JADE_FSW4_SIS_V*mm*_DIFF _V*nn*.PDF	Tracked Changes PDF between SIS version *mm* and SIS version *nn* (usually *mm* = *nn* + 1).	JADE team

5.7 EXTRAS directory

The EXTRAS directory contains files which facilitate the use of the archive volume but which are not considered part of the archive itself. Table 33 contains a list of the important contents of the EXTRAS directory. [Helpful Software may be included here, rather than a SOFTWARE directory.]

Table 33: EXTRAS subdirectory contents

File	Description	Responsibility
EXTRINFO.TXT	A description of the contents of this directory	PPI
[TBD]		

5.8 INDEX directory

The INDEX.TAB file contains a listing of all data products on the archive volume. The index (INDEX.TAB) and index information (INDXINFO.TXT) files are required by the PDS volume standards. The format of these ASCII files is described in §6.2.5. An online and web-accessible index file will be available at the PPI Node while data volumes are being produced.

Table 34: INDEX directory contents

File	Description	Responsibility
INDXINFO.TXT	A description of the contents of this directory	PPI
INDEX.LBL	A PDS detached label that describes INDEX.TAB	JSOC
INDEX.TAB	A table listing all JADE data products on this volume	JSOC

6 Archive volume format

Data that comprise the JADE standard product archives will be formatted in accordance with PDS specifications [see *Planetary Science Data Dictionary*, *PDS Archiving Guide*, and *PDS Standards Reference* in §1.9].

6.1 Volume format

Although the JADE team does not control the volume format to be used by the PDS, it is necessary to define the format in which the data sets are to be transmitted via network from the SOC to the PPI node. This will be in the form of compressed *tar* archives, as created by the open source *gtar* program. Pathnames, in lower-case letters only, will be relative to the ROOT directory, e.g., *"./data", "./index"*, etc.

6.2 File formats

The following section describes file formats for the kinds of files contained on archive volumes. For more information, see the *PDS Archive Preparation Guide* [see §1.9].

6.2.1 Document files

Document files with a `TXT` extension exist in nearly all directories. They are ASCII files with embedded PDS labels. All TXT document files contain 80-byte fixed-length records; records are terminated with a carriage return (ASCII 13) and line feed character (ASCII 10) in the 79th and 80th byte, respectively. This format allows the files to be read by many operating systems, *e.g.*, UNIX, Mac OSX, Windows, etc.

In general, documents are provided in ASCII text format. However, some documents in the `DOCUMENT` directory contain formatting and figures that cannot be rendered as ASCII text. Hence these documents are also given in additional formats such as hypertext, Microsoft Word, and Adobe Acrobat (PDF). Hypertext files contain ASCII text plus hypertext mark-up language (HTML) commands that enable them to be viewed in a web browser such as *Mozilla* or MS Internet Explorer. Hypertext documents may reference ancillary files, such as images, that are incorporated into the document by the web browser.

6.2.2 Tabular files

Tabular files (`TAB` extension) exist in the `DATA` and `INDEX` directories. Tabular files are ASCII files formatted for direct reading into database management systems on various computers. Columns are fixed length, separated by commas or white space, and character fields are enclosed in double quotation marks ("). Character fields are padded with spaces to keep quotation marks in the same columns of successive records. Character fields are left justified, and numeric fields are right justified. The "start byte" and "bytes" values listed in the labels do not include the commas between fields or the quotation marks surrounding character fields. The records are of fixed length, and the last two bytes of each record contain the ASCII carriage return and line feed characters. This line format allows a table to be treated as a fixed length record file on computers that support this file type and as a text file with embedded line delimiters on those that don't support it.

Detached PDS label files will describe all tabular files. A detached label file has the same name as the data file it describes, but with the extension `LBL`. For example, the file `INDEX.TAB` is accompanied by the detached label file `INDEX.LBL` in the same directory.

6.2.3 PDS labels

All data files in the JADE Standard Product Archive Collection have associated detached PDS labels [see the *Planetary Science Data Dictionary* and the *PDS Standards Reference* in §1.9]. These label files are named using the same prefix as the data file together with an `LBL` extension.

A PDS label provides descriptive information about the associated file. The PDS label is an object-oriented structure consisting of sets of "`keyword = value`" declarations. The object that the label refers to (*e.g.* `IMAGE`, `TABLE`, etc.) is denoted by a statement of the form:

```
^object = location
```

in which the carat character (^, also called a pointer in this context) indicates where to find the object. In a PDS label, the location denotes the name of the file containing the object, along with the starting record or byte number, if there is more than one object in the file. For example:

```
^HEADER = ("98118.TAB", 1)
^TABLE = ("98118.TAB", 1025 <BYTES>)
```

indicates that the `HEADER` object begins at record 1 and that the `TABLE` object begins at byte 1025 of the file `98118.TAB`. The file `98118.TAB` must be located in the same directory as the detached label file.

Below is a list of the possible formats for the `^object` definition in labels in this product.

```
^object      = n
^object      = n <BYTES>
^object      = "filename.ext"
^object      = ("filename.ext", n)
^object      = ("filename.ext", n <BYTES>)
```

where

- *`n`* is the starting record or byte number of the object, counting from the beginning of the file (record 1, byte 1),
- `<BYTES>` indicates that the number given is in units of bytes (the default is records),
- *`filename`* is the up-to-36-character, alphanumeric upper-case file name,
- *`ext`* is the up-to-3-character upper-case file extension,
- and all detached labels contain ASCII records that terminate with a carriage return followed by a line feed (ASCII 13_{10}, 10_{10}). This allows the files to be read by most computer operating systems, e.g., UNIX, MacOS, MSWindows, etc.

Examples of PDS labels required for the JADE archive are shown in Appendix B.

6.2.4 Catalog files

Catalog files (extension `CAT`) exist in the `Root` and `CATALOG` directories. They are plain text files formatted in an object-oriented structure consisting of sets of "`keyword = value`" declarations.

6.2.5 Index files

The PDS team provides PDS index files. The format of these files is described in this SIS document for completeness.

A PDS index table contains a listing of all data products on an archive volume. For products described by a detached PDS label, the index file points to the label file, which in turn points to the data file. A PDS index is an ASCII table composed of required columns and optional columns (user defined). When values are constant across an entire volume, it is permissible to promote the value out of the table and into the PDS label for the index table.

To facilitate users' searches of the JADE data submission, a few optional columns will be included in the index table. In particular, the file start and stop times will be included. Table 35, Table 36, Table 37 and Table 38 contains a description of the JADE archive volume index files per dataset. Index files are by definition fixed length ASCII files containing comma-delimited fields. Character strings are quoted using double quotes, and left justified in their field, followed where necessary by trailing blanks. The "Start Byte" column in Table 35 gives the location of the first byte (counting from 1) of the column within the file, skipping over delimiters and quotation marks.

Table 35: Format of index files for Level 2

Column Name	Start Byte	Bytes	Description
VOLUME_ID	2	11	Contains the value JNOJAD_nnnn, where nnnn is a 4 digit number. (See Table 10)
STANDARD_DATA_PRODUCT_ID (SID)	16	19	The "type" of the data file. (See Table 7)
DATA_SET_ID	38	32	The PDS ID of the data set of which this file is a member. (See Table 10)
PRODUCT_ID	73	27	Identifier for the product [Typically filename without version number or extension]
START_TIME	102	21	Time (UTC) of the first record in the data file.
STOP_TIME	124	21	Time (UTC) of the last record in the data file.
FILE_SPECIFICATION_NAME	147	71	The full specification name of the PDS label file (including the file name and the path) that describes the product, relative to the root of the archive volume.
PRODUCT_CREATION_TIME (or CR_DATE)	220	17	Creation time of the PDS labeled data product.
PRODUCT_LABEL_MD5CHECKSUM	239	32	Labels contain product checksums, this field records the label's checksum.

Table 36: Format of index files for Level 3

Column Name	Start Byte	Bytes	Description
VOLUME_ID	2	11	*Same description as from* Table 35
STANDARD_DATA_PRODUCT_ID (SID)	16	19	*Same description as from* Table 35
DATA_SET_ID	38	30	*Same description as from* Table 35
PRODUCT_ID	71	31	*Same description as from* Table 35
START_TIME	104	21	*Same description as from* Table 35
STOP_TIME	126	21	*Same description as from* Table 35
FILE_SPECIFICATION_NAME	149	70	*Same description as from* Table 35
PRODUCT_CREATION_TIME (or CR_DATE)	221	17	*Same description as from* Table 35
PRODUCT_LABEL_MD5CHECKSUM	240	32	*Same description as from* Table 35

Table 37: Format of index files for Level 5 binary files

Column Name	Start Byte	Bytes	Description
VOLUME_ID	2	11	*Same description as from* Table 35
STANDARD_DATA_PRODUCT_ID (SID)	16	19	*Same description as from* Table 35
DATA_SET_ID	38	30	*Same description as from* Table 35
PRODUCT_ID	71	31	*Same description as from* Table 35
START_TIME	104	21	*Same description as from* Table 35
STOP_TIME	126	21	*Same description as from* Table 35
FILE_SPECIFICATION_NAME	149	69	*Same description as from* Table 35
PRODUCT_CREATION_TIME (or CR_DATE)	220	17	*Same description as from* Table 35
PRODUCT_LABEL_MD5CHECKSUM	239	32	*Same description as from* Table 35

Table 38: Format of index files for Level 5 ASCII files

Column Name	Start Byte	Bytes	Description
VOLUME_ID	2	11	*Same description as from* Table 35
STANDARD_DATA_PRODUCT_ID (SID)	16	20	*Same description as from* Table 35
DATA_SET_ID	39	24	*Same description as from* Table 35
PRODUCT_ID	66	40	*Same description as from* Table 35
START_TIME	108	21	*Same description as from* Table 35
STOP_TIME	130	21	*Same description as from* Table 35
FILE_SPECIFICATION_NAME	153	70	*Same description as from* Table 35
PRODUCT_CREATION_TIME (or CR_DATE)	225	17	*Same description as from* Table 35
PRODUCT_LABEL_MD5CHECKSUM	244	32	*Same description as from* Table 35

6.2.6 Binary formats of files

Most JADE data files are binary, where each object of each record of a file may be encoded in a different way. Each object has a DATA_TYPE entry in the LBL (or FMT) file that describes the binary format of that object in standard PDS 3 terms. The four most common binary object types are DATE (ASCII character string of time), PC_REAL (a float), LSB_INTEGER (signed integer) and LSB_UNSIGNED_INTEGER (unsigned integer), the latter 3 being little endian.

For JADE Level 2, 3 and 5 binary records, DATE objects will always be 21 characters long in ASCII, in the PDS UTC format CCYY-DDDTHH:MM:SS.sss. LSB_INTEGER is a 'least significant byte first' (LSB, also known as little endian) signed integer and may be 1, 2 or 4 bytes long (the ITEM_BYTES or if a scalar, BYTES, entry indicates which). LSB_UNSIGNED_INTEGER is similar, but for unsigned integers. PC_REAL is an LSB float, and may be 4 or 8 bytes long (a single or double float respectively), which one is indicated by ITEM_BYTES or BYTES.

There is also a bit level value indicated by BIT_DATA_TYPE = BOOLEAN entry, either 0 or 1.

6.2.7 ASCII formats of files

Some Level 5 JADE data files are ASCII files, where each object of each record of a file may be encoded in a different way. Each object has a DATA_TYPE entry in the LBL (or FMT) file that describes the ASCII format of that object in standard PDS 3 terms. The three most common binary object types are DATE (ASCII character string of time), ASCII_REAL (a float) and ASCII_INTEGER (an integer). DATE is in the same format as given as characters in the binary files, that of the PDS UTC format CCYY-DDDTHH:MM:SS.sss.

6.2.8 Days without Data

If a given day has no data of a given type, then there is simply no file present for that day. For instance, high rate science was generally only for 6 hours per orbit, and usually on the same day; therefore the vast majority of days will have no data files for high rate science data.

6.2.9 Level 2 data files

The Level 2 data files are binary and have files ending in the extension .DAT. Accompanying them in the same directory are the label files with the same filename but the extension .LBL.

For example, the PDS file pairs will have the following paths in the Volume:

ROOT/DATA/*yyyy*/*yyyyddd*/*subdir*/JAD_L20_*aaa_bbb_ccc_yyyyddd*_V*nn*.DAT
ROOT/DATA/*yyyy*/*yyyyddd*/*subdir*/JAD_L20_*aaa_bbb_ccc_yyyyddd*_V*nn*.LBL

The format file (same filename minus the date part, but including the version number, with the extension .FMT) accompanying (and already listed within) the LBL files are usually found in the LABEL directory at the root of the volume – however it was decided to exclude this LABEL directory (and therefore exclude FMT files) as they are redundant and may be copy/pasted out of the LBL files. [FMT files are made locally for JADE file production, but do not get to the PDS.]

See section 3.1 for the explanation of JAD_L20_*aaa_bbb_ccc_yyyyddd*_V*nn*, and *subdir* is the subdirectory name given in Table 16.

To save space in this document, Table 42 gives the 25-object header for the binary files for Level 2 products, which is then used throughout. This is the same for all but a few objects, e.g. PACKETID, that gives a slightly different description for each product, where text that may be different is shown in blue.

Other objects may have similar names in different product types, i.e. MIN_SUBTRACTED_VALUE, MCP_NOT_AT_COMMANDED, SWEEP_TABLE, MCP_COMMANDED_VALUE or DATA, but may have different sizes or be different types (i.e. float or unsigned integer, of either 2 or 4 bytes, or 1 or 3 elements) depending on which Level 2 product they are.

CODMAC Level 1 data (not in PDS, see section 3.3.1) collects counts in the DATA object, however has a MIN_SUBTRACTED_VALUE removed from it prior to onboard compression. For CODMAC Level 2 data here we re-add it to the DATA object, where:

$$\text{DATA}\{\text{Level 2}\} = \text{DATA}\{\text{Level 1}\} + \text{MIN\_SUBTRACTED\_VALUE}$$

Since MIN_SUBTRACTED_VALUE is always provided, you can work out the Level 1 DATA yourself if required.

Figure 6 shows all 43 different JADE *_SCI product IDs that will go in to the PDS, grouping them together in to Science and Operations. Product IDs are numerically represented in hex, and only those with PACKETIDs greater than 0x0A (16 decimal) and less than 0xA0 (160 decimal) will go in to the PDS.

Table 40 and Table 41 summarize the type of data the 43 different JADE *_SCI products provide over 14 file types, and how they are arranged and lossy compressed. For instance 16-8 bit means that the value onboard was collected as a two-byte unsigned integer, but lossy compressed to 1 byte for transmission to ground (lossless compression may also have occurred after this step). Although low rate science can send back electron data from any individual sensor, only one is returned due to bandwidth constraints.

Table 39 lists the 14 level 2 products and provides information on how many PDS Objects are in each record, and how many bytes are in a record. The number of records per day, however, is dependent on which products are commanded and what their ACCUMULATION_TIME is (which may vary).

Note that the LBL/FMT files describe DATA as 2D containers (a container within a container that holds a scalar), but also show a 1D data array description that is commented out. The original telemetry stream is of 1D data blobs, but for convenience to the user we describe it in the 2D way in the FMT file, but you can use whichever description you find easier. The 1D ordering is based on c, in that the last dimension changes fastest, i.e. if a 1D array is x=[1, 2, 3, 4, 5, 6] and that should be a 3x2 array y, then:

y[0][0] = 1; y[0][1] = 2; y[1][0] = 3; y[1][1] = 4; y[2][0] = 5; y[2][1] = 6;

Table 39: Size of a record of each Level 2 product.

Product	Bytes per record	Objects per record
JAD_L20_ALL_ION_DER_V01	4406	27
JAD_L20_ALL_ION_DES_V01	84	35
JAD_L20_CAL_ELC_ALL_V01	13154	30
JAD_L20_CAL_ION_ANY_V01	10054	25
JAD_L20_HLC_ION_LOG_V01	3270	25
JAD_L20_HLC_ION_TOF_V01	12358	25
JAD_L20_HRS_ELC_ALL_V01	6628	32
JAD_L20_HRS_ION_ANY_V01	838	25
JAD_L20_HVE_ELC_ALL_V01	282	25
JAD_L20_HVE_ION_ALL_V01	454	25
JAD_L20_HVE_ION_LOG_V01	170	25
JAD_L20_HVE_ION_TOF_V01	582	25
JAD_L20_LRS_ELC_ANY_V01	12384	32
JAD_L20_LRS_ION_ANY_V01	10054	25

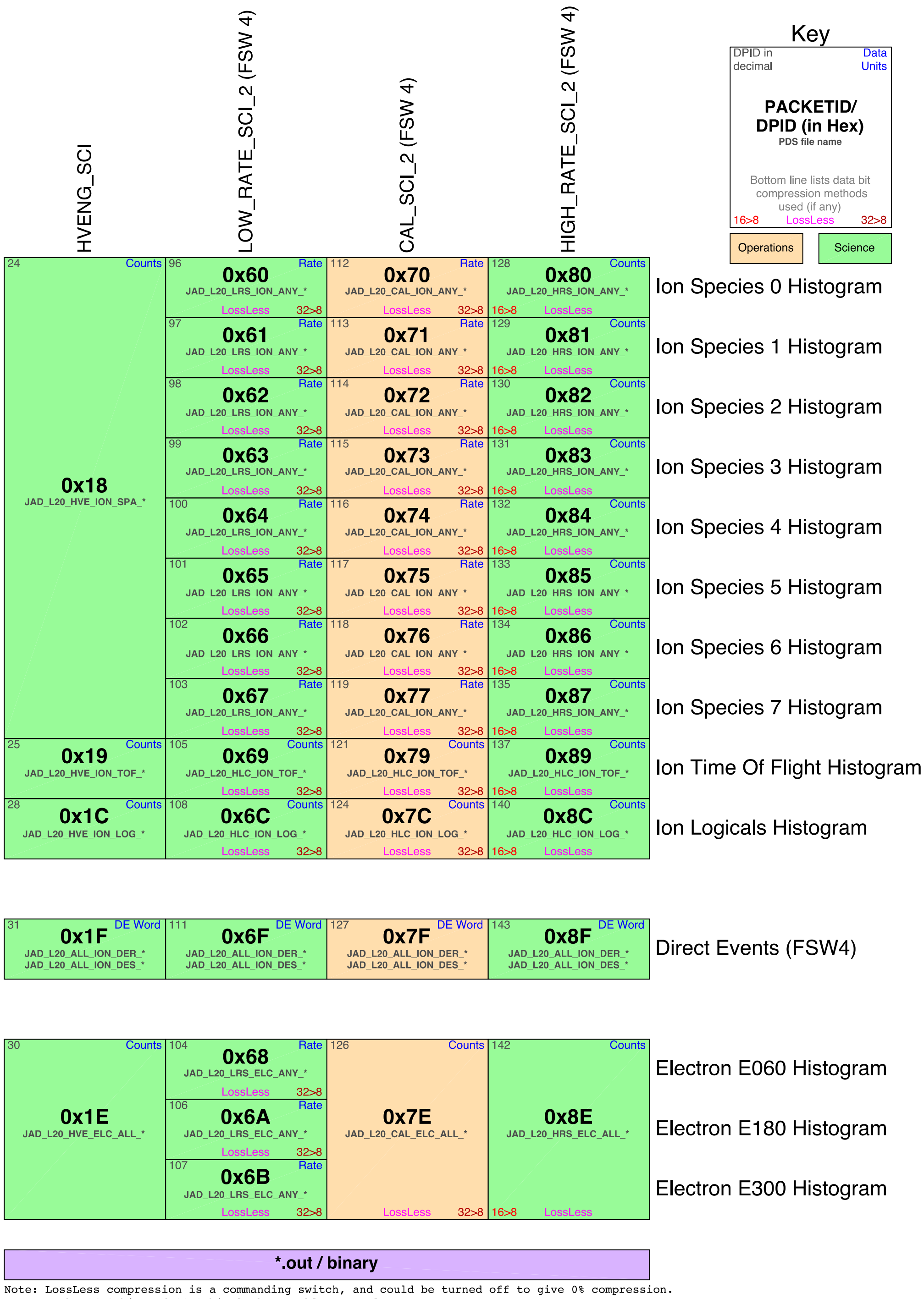


Figure 6: 'Periodic' table comparing the different JADE products, giving their packet ID number in hex (DPID in figure key, see entry in Table 42), decimal, the PDS name fragment and information on what type of compression was used, and whether it records counts per accumulation or count rates.

PDS FMT Name	TLM Spreadsheet Name	# Bytes	Packet ID / Data Type	0x8E	0x68 0x6A 0x6B	0x7E	0x1E	0x1F 0x6F 0x7F 0x8F	0x1F 0x6F 0x7F 0x8F	0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87	0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67	0x70 0x71 0x72 0x73 0x74 0x75 0x76 0x77	0x18	0x69 0x79 0x89	0x19	0x6C 0x7C 0x8C	0x1C
			Data Type	JAD_L20_HRS_ELC_ALL	JAD_L20_LRS_ELC_ANY	JAD_L20_CAL_ELC_ALL	JAD_L20_HVE_ELC_ALL	JAD_L20_ALL_ION_DER	JAD_L20_ALL_ION_DES	JAD_L20_HRS_ION_ANY	JAD_L20_LRS_ION_ANY	JAD_L20_CAL_ION_ANY	JAD_L20_HVE_ION_ALL	JAD_L20_HLC_ION_TOF	JAD_L20_HVE_ION_TOF	JAD_L20_HLC_ION_LOG	JAD_L20_HVE_ION_LOG
SYNC	Sync Pattern	4	uint32[1]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DPID_COUNT	DPID Count (Source Sequence Count)	1	uint8[1]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
COMPRESSION	Lossless Compression Status	1	uint8[1]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IDPLENGTH	IDP Length	2	uint16[1]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PACKETID	Packet ID (DPID)	1	uint8[1]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FLIGHT_OR_STL	Added on Ground (Placed to byte pack)	1	uint8[1]	^	^	^	^	^	^	^	^	^	^	^	^	^	^
PACKET_MODE	Added on Ground	1	int8[1]	^	^	^	^	^	^	^	^	^	^	^	^	^	^
PACKET_SPECIES	Added on Ground	1	int8[1]	^	^	^	^	^	^	^	^	^	^	^	^	^	^
TIMESTAMP_WHOLE	Timestamp (Whole Second)	4	uint32[1]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TIMESTAMP_SUB	Timestamp (Subsecond)	2	uint16[1]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ACCUMULATION_TIME	Accumulation Time	2	uint16[1]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TABLES_VERSION	LUT Version (float version of Hex)	4	float(1)	✓	✓	✓	^	✓	✓	✓	✓	✓	^	✓	^	✓	^
FSW_VERSION	FSW Version (float version of Hex)	4	float(1)	✓	✓	✓	^	✓	✓	✓	✓	✓	^	✓	^	✓	^
ACCUM_TRUNCATION	Accum Truncation	1	uint8[1]	✓	✓	✓	^	✓	✓	✓	✓	✓	^	✓	^	✓	^
DATA_UNITS	Added on Ground (Placed to byte pack)	1	uint8[1]	^	^	^	^	^	^	^	^	^	^	^	^	^	^
COMPRESSION_RATIO	Added on Ground	4	float[1]	^	^	^	^	^	^	^	^	^	^	^	^	^	^
UTC	Added on Ground	21	char[21]	^	^	^	^	^	^	^	^	^	^	^	^	^	^
LEAP_SECOND_VERSION	Added on Ground (Placed to byte pack)	1	uint8[1]	^	^	^	^	^	^	^	^	^	^	^	^	^	^
SCLKSCET_VERSION	Added on Ground	2	int16[1]	^	^	^	^	^	^	^	^	^	^	^	^	^	^
ISSUES (bit flags)	Added on Ground	4	uint32[1]	^	^	^	^	^	^	^	^	^	^	^	^	^	^
MIN_SUBTRACTED_VALUE	Min Subtracted Value	4	uint32[1]	✓		✓	^	^	^	✓			^	✓	^	✓	^
	Min Subtracted Value	4	float(1)		✓						✓	✓					
MCP_NOT_AT_COMMANDED	MCP not at Commanded	1	uint8[1]		✓			✓	✓	✓	✓	✓	^	✓	^	✓	^
SWEEP_TABLE	Sweep Table	1	uint8[1]		✓			✓	✓	✓	✓	✓	^	✓	^	✓	^
MCP_COMMANDED_VALUE	MCP Commanded, or, Ion MCP Commanded	2	uint16[1]		✓			✓	✓	✓	✓	✓	^	✓	^	✓	^
MCP_NOT_AT_COMMANDED	E060, E180 and E300 MCPs not at Commanded	3	uint8[3]	✓		✓	^										
SWEEP_TABLE	E060, E180 & E300 Sweep Tables	3	uint8[3]	✓		✓	^										
MCP_COMMANDED_VALUE	E060, E180 & E300 MCPs Commanded	6	uint16[3]	✓		✓	^										
	Array Data Blob	Depends	uint16	✓						✓							
DATA	Array Data Blob	Depends	uint32			✓	✓						✓	✓	✓	✓	✓
	Array Data Blob	Depends	float		✓						✓	✓					
MAG_TIME_DIR	mag_time_direction	1	uint8[1]	✓													
MAG_LOOK_DIR	mag_anode	1	uint8[1]	✓													
MAG_FIRST_DFL	first_dfl	2	uint16[1]	✓													
MAG_ELEVATION	elevation_sign + elevation_angle	1	int8[1]	✓													
MAG_TIME_LATENCY	mag_time_latency	1	uint8[1]	✓													
MAG_TIMESTAMP_WHOLE	mag_whole_seconds or Mag_time_whole	4	uint32[1]	✓	✓	✓											
MAG_TIMESTAMP_SUB	Mag_time_sub	2	uint16[1]		✓	✓											
MAG_COUNT_VALID	Mag_count_valid	1	uint8[1]		✓	✓											
MAG_COUNT_INVALID	Mag_count_invalid	1	uint8[1]		✓	✓											
MAG_VECTOR	mag_x+mag_y+mag_z	12	int32[3]	✓	✓	✓											
BACKGROUND_COUNTS	Background Counter	4	uint32[1]	Inc. in DATA	✓	Inc. in DATA	Inc. in DATA										
ESENSOR	Added on Ground	2	uint16[1]		^												
DE_COL_SUB_SEQ_COUNT	Collection sub-sequence count	2	uint16[1]					✓	✓								
DATA[*]	*DataBlob Padded to Fixed Size*	*4332*	uint16[2166]					✓									
DE_SIZE	Added on Ground	2	uint16[1]					^									
DE_BAD	Added on Ground	1	uint8[1]						^								
DE_EVENT0_BOUNDARY1_MARKER2	Added on Ground	1	uint8[1]						^								
DE_SWEEP_NUMBER	Added on Ground	2	uint16[1]						^								
DE_ESA_STEP	Added on Ground	1	uint8[1]						^								
DE_DFL_STEP	Added on Ground	1	uint8[1]						^								
DE_TOF	Added on Ground	2	uint16[1]						^								
DE_ANODE_ID	Added on Ground	1	uint8[1]						^								
DE_QUALITY_FLAG_2	Added on Ground	1	uint8[1]						^								
DE_QUALITY_FLAG_1	Added on Ground	1	uint8[1]						^								
DE_QUALITY_FLAG_0	Added on Ground	1	uint8[1]						^								

Figure 7: Breaking out the JADE Level 2 products in to the different Objects to allow similarities to be drawn.
There are a total of 14 products, compressed here for readability. A green shaded ^ mark values added on the ground, red shaded ones means MISSING_CONSTANTS were added. Electron products include a background anode; only LRS data splits it out from the DATA object.

Table 40: Data Collection types by dimensions.
One spin is 48 E-Spin-Phase Sectors or 78 I-Spin-Phase Sectors.
Ion species may be commanded to return 1-8 species for HRS/LRS/CAL modes, but 3 is typical.
Electron background anodes have been ignored for this table.
A full ion energy sweep is of 64 steps, however each packet contains either the top 32 or bottom 32 only.

	HRS	LRS	MCP CAL	HVE
Electrons	3 Sensors 64 Energies 48 Anodes *(Same as CAL)*	1 Sensor 64 Energies 48 E-Spin-Phase	3 Sensors 64 Energies 48 Anodes *(Same as HRS)*	3 Sensors 1 Energy 48 Anodes
Ion Species	1-8 Species 32 Energies 12 Anodes	1-8 Species 32 Energies 78 I-Spin-Phase		8 Species 1 Energy 12 Anodes
Ion TOF	32 Energies 96 TOF			1 Energy 128 TOF
Ion Logicals	32 Energies 25 Logs			1 Energy 25 Logs
Ion DE	DE Words			

Table 41: Data Collection types by units (green), lossy bit compression (red) and number of Level 2 files (purple).

	HRS	LRS	MCP CAL	HVE
Electrons	1 File Counts 16>8 bit	3 Files *Rate* 32>8 bit	1 File Counts 32>8 bit	1 File Counts None
Ion Species	1 File Counts 16>8 bit	1 File *Rate* 32>8 bit	1 File *Rate* 32>8 bit	1 File Counts None
Ion TOF	1 File Counts 16>8 bit	1 File Counts 32>8 bit	1 File Counts 32>8 bit	1 File Counts None
Ion Log	1 File Counts 16>8 bit	1 File Counts 32>8 bit	1 File Counts 32>8 bit	1 File Counts None
Ion DE	1 File DE Word None	1 File DE Word None	1 File DE Word None	1 File DE Word None

Counts = Total Counts,
Rate = Count rates (normalized by number of views)

The following table (over 4 pages) describes the header that is identical for all the following data products. The names and word type (int/float/etc.) for all level 2 data is also summarized in Figure 7. Any text in *red italics* is a note that is not in the LBL nor FMT file, while any text in **blue boldface** may change depending on the product (usually just the product ID or species number). This color system will apply for format tables throughout the rest of section 6.2.

Table 42: Format of Level 2 data record header for all binary data files.

Byte	Bit	Length (bits)	Name	Description
1	0	32	SYNC	JADE Sync Pattern for IDP packets. Hex value = 0xFAF33403, Decimal = 4210242563
5	0	8	DPID_COUNT	DPID Count (Source Sequence Count) Count of the number of times this product has been generated since the startup (or reset) of the generating application (Boot Program or Science Program). This count resets to 0 upon entry to the modes of BOOT, LVENG, HVENG, LOW_RATE_SCI, MCP_CAL_SCI, HI_RATE_SCI, LOW_RATE_SCI2, MCP_CAL_SCI2, HI_RATE_SCI2. Note: starts with 0, increments by 1, eventually rolls over at 255.
6	0	8	COMPRESSION	Lossless Compression Status. Indicates whether the data (non-header) segment of the IDP packet (IDP Data) was lossless compressed. 0 = Not Compressed **1 = Compressed** *Last line only shown if the packet could be compressed.*
7	0	16	IDPLENGTH	IDP Length, Byte Length of the IDP packet. Uncompressed size for this product should be **416**.
9	0	8	PACKETID	Packet ID (DPID), Data Product Identifier *Followed by Name of Packet ID for each product, e.g.* **High Rate Science - Ion Species Histogram** **Each packet is one of the following ion species:** **SP0, species 0, PACKETID = 128 /* 0x80 */** **SP1, species 1, PACKETID = 129 /* 0x81 */** **SP2, species 2, PACKETID = 130 /* 0x82 */** **SP3, species 3, PACKETID = 131 /* 0x83 */** **SP4, species 4, PACKETID = 132 /* 0x84 */** **SP5, species 5, PACKETID = 133 /* 0x85 */** **SP6, species 6, PACKETID = 134 /* 0x86 */** **SP7, species 7, PACKETID = 135 /* 0x87 */**
10	0	8	FLIGHT_OR_STL	In Flight data, or STL (ground EM tests): 0 = In flight, from JADE on Juno (via FEI) 1 = On ground, from STL tests (via FEI) 2 = On ground, from SwRI tests (not FEI) 255 = Unknown

Byte	Bit	Length (bits)	Name	Description
11	0	8	PACKET_MODE	Packet Mode, describes type of data telemetry. -2 = HSK / Housekeeping Engineering -1 = HVE / High Voltage Engineering 0 = CAL / MCP Calibration Science 1 = LRS / Low Rate Science 2 = HRS / High Rate Science 127 = Unknown 254 = Wrong - but HSK, see below. 255 = Wrong - but HVE, see below. (Note, this could also be calculated via PACKETID.) If you have 254 or 255 then your code is incorrect, check you read a signed byte, rather than unsigned.
12	0	8	PACKET_SPECIES	Packet Species, describes type of plasma data. -1 = electrons 0 = ion species 0, SP0 1 = ion species 1, SP1 2 = ion species 2, SP2 3 = ion species 3, SP3 4 = ion species 4, SP4 5 = ion species 5, SP5 6 = ion species 6, SP6 7 = ion species 7, SP7 8 = Not Used 9 = All ions 127 = Unknown 255 = Wrong - but electrons, see below. If you have 255 then your code is incorrect, check you read a signed byte, rather than unsigned.
13	0	32	TIMESTAMP_WHOLE	Timestamp (Whole Second). Timestamp (whole second) of the data for this packet when collection began. This is sometimes referred to as Mission Elapsed Time (MET) and is Referenced from 2000-001T12:00:00.000 UTC, but 1 tick is not exactly 1 S.I. second. See UTC object for corrected converted time. Note: Spacecraft Clock = TIMESTAMP_WHOLE:TIMESTAMP_SUB
17	0	16	TIMESTAMP_SUB	Timestamp (Subsecond). Timestamp subsecond of the data for this packet when collection began. Unit: Microseconds scaled to 16 bits. Note: Spacecraft Clock = TIMESTAMP_WHOLE:TIMESTAMP_SUB

Byte	Bit	Length (bits)	Name	Description
19	0	16	ACCUMULATION_TIME	Accumulation Time. Number of seconds over which the data in this product was collected (Science Program). *NOTE: This is not the collection period of a given energy step at a given angle, etc., that would be used to convert to counts/second. It is the time it took to collect all the data measured within this record.*
21	0	32	TABLES_VERSION	Look Up Tables (LUT) version used onboard. All tables are combined (compression, sweeping, macros, etc.) onboard in to a large image. This is the image number, or table version. Number should be to 2 decimal places.
25	0	32	FSW_VERSION	Flight Software version used. Number should be to 2 decimal places.
29	0	8	ACCUM_TRUNCATION	Accumulation Truncation, Whether commanded accumulation time ended early. 0 = Nominal 1 = Early 255 = Unknown
30	0	8	DATA_UNITS	Science Data could be total counts (per accumulation) or a rate, normalized to counts per view. 0 = All counts in the accumulation period (int) 1 = All counts divided by number of views (float) 2 = Counts per second (float) 255 = Not appropriate for this dataset, or Unknown.
31	0	32	COMPRESSION_RATIO	Data compression ratio of data blob when it was transmitted to Earth: Ratio = {Uncompressed size}/{Compressed size} This is the compression due to the lossless scheme, and does not include any lossy compression which may have occurred prior to it, such as the 32-bit to 8-bit or 16-bit to 8-bit look up tables that are often used prior to the lossy compression. A value of 1 means there was no lossless data compression, i.e. it was turned off, and object COMPRESSION should equal 1.
35	0	168	UTC	UTC timestamp, of format yyyy-dddTHH:MM:SS.sss where yyyy = year, ddd = day of year, HH = hour, MM = minute, SS.sss = decimal seconds to millisecond resolution. Value calculated via SPICE from spacecraft clock time, {TIMESTAMP_WHOLE}:{TIMESTAMP_SUB}
56	0	8	LEAP_SECOND_VERSION	The NAIF SPICE kernel for lsk used to generate UTC. The lsk (leap second kernels) files are used in time conversions and have filenames naifnnnn.tls, where is the lsk version number (with leading zeros).

Byte	Bit	Length (bits)	Name	Description
57	0	16	SCLKSCET_VERSION	The NAIF SPICE kernel for sclk used to generate UTC. The JUNO sclk files are used to convert the spacecraft clock timestamps to UTC time, and all have filenames JNO_SCLKSCET.nnnnn.tsc, where nnnnn is the SCLKSCET version number (with leading zeros and positive). Each kernel has a reconstructed and predicted part for it's values, typically any time after the last row of the SCLK01_COEFFICIENTS_61999 table is predicted. If TIMESTAMP_WHOLE:TIMESTAMP_SUB is in the predicted part then SCLKSCET_VERSION will be negative, the absolute value would be the version number. If a later SCLKSCET kernel version is used the UTC time will likely be different. If TIMESTAMP_WHOLE:TIMESTAMP_SUB is in the reconstructed region the number will be positive (equal to the version number) and will not vary with later kernels. e.g. If SCLKSCET_VERSION = -17 then kernel JNO_SCLKSCET.00017.tsc was used to convert to UTC, but it's a predicted UTC time. If SCLKSCET_VERSION = 18 then kernel JNO_SCLKSCET.00018.tsc was used to convert to UTC, and it's a reconstructed UTC time that will not change with later SCLKSCET kernel versions. Within the PDS archive this value should always be positive.
59	0	32	ISSUES	*The ISSUES description is far too long to fit in this table, see* Table 45 *instead.*
63	0	32	MIN_SUBTRACTED_VALUE	Minimun Subtracted Value. Minimum value subtracted from every element in the array data blob for transmission to Earth. (This has already been added back to the DATA.) *If DATA_UNITS = 1, this is followed by:* **Note: the units are rates (counts per views), are floats rather than integers, and are fractions of 1/512.**
67	0	8 or 24	MCP_NOT_AT_COMMANDED	*See* Table 43 *and* Table 44 *for details of these final three header objects.* Table 43 *for ions and LRS electrons,* *or* Table 44 *for other electron products.*
68 or 70	0	8 or 24	SWEEP_TABLE	
69 or 73	0	16 or 48	MCP_COMMANDED_VALUE	

Table 43: Format of Level 2 data record sub-header for all binary data files of one sensor.

Byte	Bit	Length (bits)	Name	Description
67	0	8	MCP_NOT_AT_ COMMANDED	MCP not at Commanded flag. This denotes whether the MCP voltage was reduced during the data collection. 0 = Nominal 1 = Reduced 255 = Unknown
68	0	8	SWEEP_TABLE	Which sweep table does the **ion** sensor have, **0-3**.
69	0	16	MCP_COMMA NDED_VALUE	**Ion** MCPs Commanded raw DAC value.

Table 44: Format of Level 2 data record sub-header for all binary data files of all electron sensors.

Byte	Bit	Length (bits)	Name	Description
67	0	24	MCP_NOT_AT_ COMMANDED	MCP not at Commanded flag. This denotes whether the MCP voltage was reduced during the data collection. (The three values are for the MCPs of E060, E180 and E300 respectively.) 0 = Nominal 1 = Reduced 255 = Unknown
70	0	24	SWEEP_TABLE	Which sweep table do the electron sensors have, 0-2. (The three values are for the MCPs of E060, E180 and E300 respectively.)
73	0	48	MCP_COMMA NDED_VALUE	Electron MCPs Commanded raw DAC value. (The three values are for the MCPs of E060, E180 and E300 respectively.)

The ISSUES object description is far too large to fit in the tables above, so is given here in two column format and a reduced font size over two pages (with some line breaks from the FMT/LBL file contents altered/removed). Reading the FMT/LBL file itself may be clearer.

Table 45: Full description of the ISSUE flag in the FMT/LBL files.

Issues or potential issues in this data record.
These are issues that can be identified within the JADE packet of data itself without any external information. e.g. timing issues due to the MAG time stutter, or any voltage pulsing, would not be included as there are no indicators to them within this JADE packet.
[For a more comprehensive list of potential issues from internal and external sources please see the Level 3 data.]

Level 2 issues of this JADE packet are flagged by individual bits, and several may be hit. If no issues are flagged then this 4-byte unsigned integer is zero. A value of 4294967295 is the MISSING_CONSTANT and means that the issue status is currently unknown.

All bits at 0 implies all is okay as seen by this packet. If a bit is set to 1 then that bit is flagged, otherwise it is set to zero and unflagged.

The bits are set as followed, grouped in to seriousness:

Not very serious issues for doing science:
Bit 0 = UTC time is predicted, yet to be finalized.
Bit 1 = Position/Orientation values predicted, yet to be finalized. Level 3 (and above) data only.
Bit 2 = TABLES_VERSION object was altered on the ground to accurately reflect a 'commanded parameter update' outside the initial per-orbit commands JADE is returning.
[If changed, the original downlinked TABLES_VERSION value can be found by cross-referencing the PARAM_TABLE_VER object in the JAD_L20_HSK_ALL_SHK files. Note here the PARAM_TABLE_VER value is given as a unsigned integer of Hex Major-Middle-Minor, such that a value of 770 decimal is in hex 0x302, meaning Table Version 3.02]
Bit 3 = FSW_VERSION 4.00 LRS/CAL Ion Species bug fixed on the ground by adjusting TIMESTAMP_WHOLE, TIMESTAMP_SUB, and ACCUMULATION_TIME based on cross-referencing JADE commanding.
Bit 4 = LRS/CAL Ion Species record with unobserved look directions (views) populated using views from neighboring record. See Bit 12 for uncorrected/unpopulated description.
(Only possible if ACCUMULATION_TIME = 30.)
Bit 5 = TIMESTAMP_WHOLE/SUB adjusted on the ground to mitigate any Juno time stutter effects.
[Other TIMESTAMPs are susceptible to the onboard time stutter too, but only the JADE packet TIMESTAMP_WHOLE/SUB is tracked here.]
Bit 6 = Currently unused.
Bit 7 = Warning, a leap second occurs during the accumulation period.

Data slightly different than expected, but can be used for science with a little extra coding:
Bit 8 = ACCUM_TRUNCATION object flagged.
Bit 9 = Electron (HRS/LRS/CAL) MAG objects are not tracked, are either zeros or MISSING_CONSTANT.
[LRS and CAL did not have MAG objects prior to FSW_VERSION 4.10, therefore those MAG objects here are set to MISSING_CONSTANT when FSW_VERSION < 4.10.]
Bit 10 = TIMESTAMP_WHOLE/SUB affected by a Juno onboard time stutter, JADE reported timestamp is likely 1 whole tick too large.
[Other TIMESTAMPs are susceptible to the onboard time stutter too, but only the JADE packet TIMESTAMP_WHOLE/SUB is tracked here.]
Bit 11 = Currently unused.
Bit 12 = LRS/CAL Ion Species record potentially has unobserved look directions (spin phase sectors or views) present in the data, meaning the record may not contain data for a full 4pi steradians field-of-view.
Unobserved look directions have zero counts per view (or counts per second) in the data, although an observed look direction may also have zero counts if no ions were measured. Therefore there is a potential confusion over zero measured counts or simply unmeasured. e.g. if the spin period is 30.7 seconds, then not all of the 78 spin phase sectors will be sampled in 30 seconds. (Unobserved views are only possible if ACCUMULATION_TIME <= 30.) See the JADE SIS for more information.
Bit 13 = At least one anode is blanked.
See SIS document for further information.
Bit 14 = FSW_VERSION 4.00 LRS/CAL Ion Species bug warning:
Not fixed as yet - when fixed it will become bit 3 of ISSUES instead.
Level 2 data only when FSW_VERSION = 4.00, ACCUMULTION_TIME object is MISSING_CONSTANT.
Also, TIMESTAMP_WHOLE:TIMESTAMP_SUB is the end of the packet rather than the usual start, see TIMESTAMP_WHOLE object for more details.
[Only affects data from 2015-089 to 2015-115.]
Bit 15 = Electron Anodes Reversed.
Level 2 data only when FSW_VERSION < 4.10 and only electron packets. Electron anodes are reversed in order and need to be remapped, however electron Spin Phase data (LRS data) cannot be remapped. See the SIS document for more information about this.
[Affects all electron data 2011 to 2015-115.]

Data very different than expected, may not be suitable for science - use with extreme caution.

Bit 16 = Data is not from flight instrument on Juno, see FLIGHT_OR_STL object.

Bit 17 = MCP_NOT_AT_COMMANDED object flagged. Electron HRS/CAL/HVE packets use all three electron sensors and therefore have three MCP_NOT_AT_COMMANDED values per packet. Setting this flag means at least one of those three mcps is not at its commanded value.

Bit 18 = Data includes some JADE-E300 sensor data. (Only flagged for HRS, LRS, CAL and HVE data.) E300 has a high voltage power supply issue and reported energy steps may be incorrect. If E300 is off but still reported in the data product, it may be zeros of fill values.

Bit 19 = Ion packet abruptly truncated. This packet should not be used. It had an ACCUMULATION_TIME = 1, ACCUM_TRUNCATION = 1 and the DATA object is all zeros, with a timestamp that matches an earlier valid packet that was not truncated and has non-zero DATA. e.g. TOF and LOG example in level 2 data at TIMESTAMP_WHOLE of 495879710 (UTC 2015-261).

Bit 20 = MCP Dipping Triggered, in one or more sensors. If the sensor measures excessive counts, it temporarily lowers the MCP voltage to reduce the number of counts and protect the sensor. The MCP_NOT_AT_COMMANDED object is also flagged (Bit 17 in ISSUES) since the MCP is no longer at the commanded voltage. For HRS/CAL/HVE electrons (datasets where multiple sensors are on) it is possible that one sensor has been dipped, but the others are not and still providing good data. (First MCP dip was HRS electrons, 2017-350.)

Bit 21 = MCP Dipped sensor's DATA set to fill values. If MCP dipping has triggered (Bit 20 of ISSUES) then: DATA and BACKGROUND objects (and their *_SIGMAs) have been replaced with MISSING_CONSTANT values. (Never used for Level 2 data, which has the counts as measured in the dipped state.) In addition, Bit 17 of the ISSUES object (i.e. MCP_NOT_AT_COMMANDED object = 1) is set to zero, and, if it exists, the MCP_NOT_AT_COMMANDED object itself is changed (from 1) to be 0 for the offending sensor(s). If the DATA object contains data from multiple sensors (HRS/CAL/HVE electrons) then only the elements of the DATA object for the dipped sensor are set to MISSING_CONSTANT (as identified by the MCP_NOT_AT_COMMANDED value for each sensor (prior to setting them to 0)). [See Bit 22 for a similar flag.]

Bit 22 = 1 or more ELC sensor DATA set to fill values. Affects only electron HRS/CAL/HVE products (i.e. products that use multiple sensors), and generally only when starting that mode. When switching to HRS/CAL/HVE from LRS, one JADE-E sensor is already on, and the other(s) have to turn on, then it takes some time for that sensor to reach the commanded voltage. For a given record, MCP_NOT_AT_COMMANDED = 0 for one sensor but is still = 1 for others. That is one sensor is taking valid science but the other(s) are not there yet and for those sensors: DATA and BACKGROUND objects (and their *_SIGMAs) have been replaced with MISSING_CONSTANT values. (Never used for Level 2 data, which has the counts as measured in the dipped state.) In addition, Bit 17 of the ISSUES object (i.e. MCP_NOT_AT_COMMANDED object = 1) is set to zero, and, if it exists, the MCP_NOT_AT_COMMANDED object itself is changed (from 1) to be 0 for the offending sensor(s). Only the elements of the DATA object for the original MCP_NOT_AT_COMMANDED = 1 sensor(s) (prior to setting them to 0) are set to MISSING_CONSTANT. [Bits 21 and 22 are essentially the same feature caused by an mcp voltage not being at the commanded value, but the reason why this is the case is different. The treatment is identical for both Bit 21 and Bit 22.]

Bit 23 = Currently unused.

Bit 24 = Currently unused.
Bit 25 = Currently unused.
Bit 26 = Currently unused.
Bit 27 = Currently unused.
Bit 28 = Currently unused.
Bit 29 = Currently unused.
Bit 30 = Currently unused.
Bit 31 = Reserved for MISSING_CONSTANT use.

Each bit has a decimal value of 2^{bit number}, and the Issues flag is the sum of 2^{flagged bit numbers}. For instance, if this ISSUES flag = 131329, then in binary that value is 00000000000000100000000100000001 showing bits 17, 8 and 0 are flagged.

[If a currently unused bit is set, please check the latest LBL file for this product that you can find to see if it now has a definition.]

6.2.9.1 Electron Data

JADE-E consists of three electron sensors. For high rate science, calibration and high voltage engineering; data from all three sensors are returned in each record. For low rate science, only one sensor of the three is on at any given time, therefore just that one sensor's data is returned per record. Each electron sensor has 16 anodes and 1 background anode. The following figure provides the look directions and numbering of the different anodes (excluding background ones) from 0 to 47.

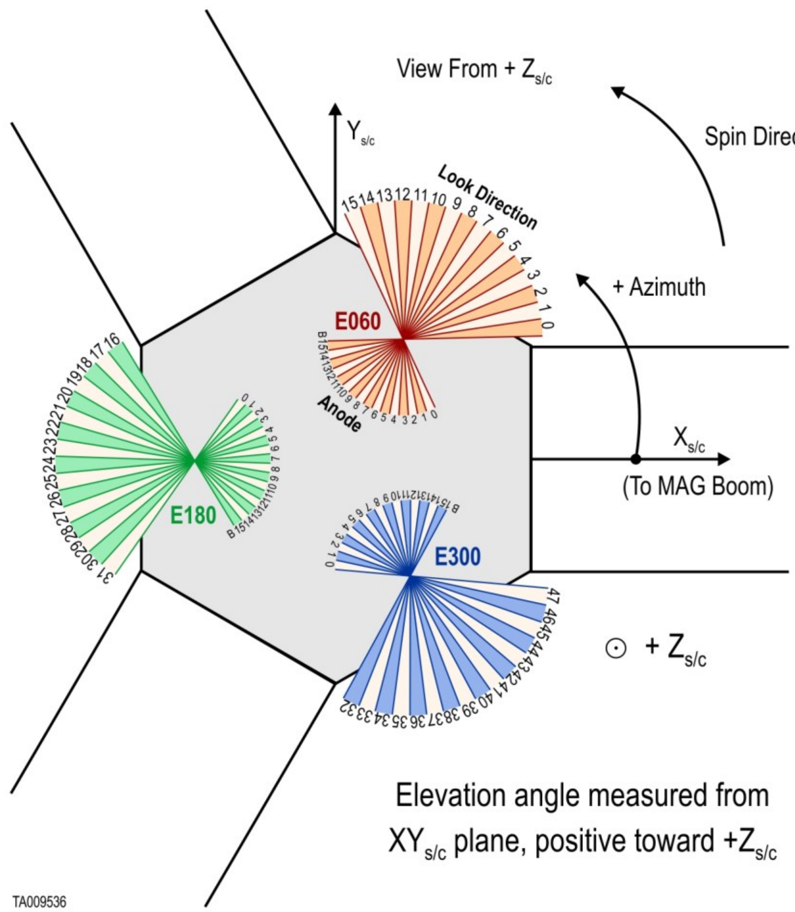


Figure 8: JADE-E Anodes vs. Look Directions (see Table 46 for the values).

For FSW4.00 only (or any FSW3) with anodes reversed (see ISSUES flag), you need to reorder the data packets of 51 bins to get sequential azimuth coverage, using this anode order:

```
{15, 14, 13, 12, 11, 10,  9,  8,  7,  6,  5,  4,  3,  2,  1,  0,  /* E060 */
 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16,  /* E180 */
 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32,  /* E300 */
 48, 49, 50  /* Background anodes remain at end*/ };
```

Table 46 maps bins to the sensor anodes and look directions for CAL, HRS, HVE and LRS, and should be used as the guide to convert to field of view directions for level 3 data. LRS data arrays are returned in electron spin phase sectors, calculated from *look_direction_id* as shown in section 6.2.9.1.4.

Table 46: Electron anode mapping to bins and look directions. (Note: Electron spin phase sectors are over many anodes and require an equation, see section 6.2.9.1.4.)

Sensor	Hardware Anode (*anode_id* or *a_id*)	JADE SC Look Direction	JSIB Data Index	HRS/CAL/HVE Bin	LRS Sensor Loop Index (*look_direction_id*)
E060	*15*	*0*	*15*	0	0
E060	*14*	*1*	*14*	1	1
E060	*13*	*2*	*13*	2	2
E060	*12*	*3*	*12*	3	3
E060	*11*	*4*	*11*	4	4
E060	*10*	*5*	*10*	5	5
E060	*9*	*6*	*9*	6	6
E060	*8*	*7*	*8*	7	7
E060	*7*	*8*	*7*	8	8
E060	*6*	*9*	*6*	9	9
E060	*5*	*10*	*5*	10	10
E060	*4*	*11*	*4*	11	11
E060	*3*	*12*	*3*	12	12
E060	*2*	*13*	*2*	13	13
E060	*1*	*14*	*1*	14	14
E060	*0*	*15*	*0*	15	15
E180	*15*	*16*	*31*	16	0
E180	*14*	*17*	*30*	17	1
E180	*13*	*18*	*29*	18	2
E180	*12*	*19*	*28*	19	3
E180	*11*	*20*	*27*	20	4
E180	*10*	*21*	*26*	21	5
E180	*9*	*22*	*25*	22	6
E180	*8*	*23*	*24*	23	7
E180	*7*	*24*	*23*	24	8
E180	*6*	*25*	*22*	25	9
E180	*5*	*26*	*21*	26	10
E180	*4*	*27*	*20*	27	11
E180	*3*	*28*	*19*	28	12
E180	*2*	*29*	*18*	29	13
E180	*1*	*30*	*17*	30	14
E180	*0*	*31*	*16*	31	15
E300	*15*	*32*	*47*	32	0
E300	*14*	*33*	*46*	33	1
E300	*13*	*34*	*45*	34	2
E300	*12*	*35*	*44*	35	3
E300	*11*	*36*	*43*	36	4
E300	*10*	*37*	*42*	37	5
E300	*9*	*38*	*41*	38	6
E300	*8*	*39*	*40*	39	7
E300	*7*	*40*	*39*	40	8
E300	*6*	*41*	*38*	41	9
E300	*5*	*42*	*37*	42	10
E300	*4*	*43*	*36*	43	11
E300	*3*	*44*	*35*	44	12
E300	*2*	*45*	*34*	45	13
E300	*1*	*46*	*33*	46	14
E300	*0*	*47*	*32*	47	15
E060	*Background*	*N/A*	*48*	48	N/A
E180	*Background*	*N/A*	*49*	49	N/A
E300	*Background*	*N/A*	*50*	50	N/A

The electron products may all contain the onboard MAG vector, depending if the magnetic field magnitude is above a threshold value, set in the uploaded Look-Up Tables (LUT, also known as the TABLES_VERSION object in Level 2 files). From LUT 3.00 (2015, start of this archive) the threshold magnitude was set to 200 nT, meaning that a valid MAG_VECTOR object was only returned near perijove (and never in the solar wind) when the magnetic field magnitude was stronger than 200 nT. This changed in LUT 3.11 (starting 2017-074) to 25 nT. Future LUT versions may change this threshold again, generally there is a usable MAG_VECTOR (when the threshold is met) or MAG_VECTOR = [0,0,0], indicating that the threshold was not met, and the true mag vector is unknown to the JADE instrument. (Earlier LUTs (pre 2015) had also set the threshold to zero for operational testing.)

This MAG_VECTOR object (when the threshold is met and not returning zeroes) is only a guide. The user should cross reference the JADE timestamp with the Level 3 MAG (also known as FGM) team's calibrated magnetic field vectors for any publications involving magnetometer data.

JADE-E has three electron sensors, but publications may refer to only two electron sensors. This is because E300 was turned off in 2016 (prior to arrival at Jupiter) and remains off. As such, any data product still including E300 data (such as JAD_L20_HRS_ELC_ALL_*) will have elements relating to E300 populated with the MISSING_CONSTANT value (see Table 48).

6.2.9.1.1 JAD_L20_CAL_ELC_ALL_*

The electron product for calibration mode is PACKETID 0x7E and includes data from all three electron sensors.
The DATA object is 2-D, 64 energies x 51 anodes, and is described in Table 47.

From 2016 onwards E300 will remain off, but the downlinked product still contains values from all three sensors. For bins that are from E300 (32 to 47 and 50), their values will be replaced with the DATA object's MISSING_CONSTANT value, so should be obvious.

*Table 47: Format of Level 2 data records for JAD_L20_CAL_ELC_ALL_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 44 *for bytes 1 to 78.*				
79	0	104448	DATA	DATA: Electron Counts 64 Energies x 51 Bins The 51 Bins are: 0 - 15 : E060 Look Directions 0-15 16 - 31 : E180 Look Directions 0-15 32 - 47 : E300 Look Directions 0-15 48 : E060 Background Anode 49 : E180 Background Anode 50 : E300 Background Anode (See SIS document for a figure.)
13135	0	32	MAG_TIMESTAMP_WHOLE	MAG_TIMESTAMP_WHOLE Whole-second timestamp of last received MAG vector *before* data collection start. Referenced from 12:00UTC 2000/01/01. [May be affected by a Juno Time Stutter.]
13139	0	16	MAG_TIMESTAMP_SUB	MAG_TIMESTAMP_SUB Sub-second timestamp of last received MAG vector *before* data collection start. A value of 65535 could be real or a MISSING_CONSTANT, however it is MISSING_CONSTANT only if MAG_TIMESTAMP_WHOLE = 0, e.g. WHOLE and SUB must both be real or both be MISSING_CONSTANT. Unit: Microseconds scaled to 16 bits. [May be affected by a Juno Time Stutter.]
13141	0	8	MAG_COUNT_VALID	MAG_COUNT_VALID Count of valid (above threshold and not saturated) MAG vectors between start of *previous* packet and start of this packet. Note: This saturates at 255. e.g if there is a 600s accumulation period, and the MAG vector is given every 2-seconds, then that's 300 counts. If all are valid then that 300 will be expressed as 255, however MAG_COUNT_INVALID would still be zero.

Byte	Bit	Length (bits)	Name	Description
13142	0	8	MAG_COUNT_ INVALID	MAG_COUNT_INVALID Count of invalid (below threshold or saturated) MAG vectors between start of *previous* packet and start of this packet. Note: This saturates at 255. e.g if there is a 600s accumulation period, and the MAG vector is given every 2-seconds, then that's 300 counts. If all are invalid then that 300 will be expressed as 255, however MAG_COUNT_VALID would still be zero.
13143	0	96	MAG_VECTO R	Last received MAG vector in nT before data collection start: 3 components [X, Y, Z] MAG range is +/- 16 G, hence limits. The coordinate system is spacecraft based, with: +X is between E060 and E300, along the 0 degree mark where E060 anode 0 starts +Y is 90 degrees, between E060 anodes 11 and 12, +Z is the spin axis. Note: these are signed integers. A Mag vector of [0 0 0] has four meanings: [Meanings 1 and 2 require MAG_TIMESTAMP_WHOLE = 0. The MAG_COUNT_VALID and MAG_COUNT_INVALID objects can help distinguish meaning 1 from 2.] 1) JADE never received a mag vector at all. (So initialized to 0s.) e.g. MAG_COUNT_VALID = 0 for this record. 2) A 25s timeout has expired without JADE receiving a MAG vector over a threshold magnitude. [Meanings 3 and 4 require MAG_TIMESTAMP_WHOLE > 0] 3) The threshold parameter was set to 0 nT. (Some early HVCO1 check-out data may have this.) 4) The broadcast message was corrupted and the magnitude and components mismatched.

6.2.9.1.2 JAD_L20_HRS_ELC_ALL_*

The electron product for high rate science is PACKETID 0x8E and includes data from all three electron sensors.
The DATA object is 2-D, 64 energies x 51 anodes, and is described in Table 48.

From 2016 onwards E300 will remain off, but the downlinked product still contains values from all three sensors. For bins that are from E300 (32 to 47 and 50), their values will be replaced with the DATA object's MISSING_CONSTANT value, so should be obvious.

*Table 48: Format of Level 2 data records for JAD_L20_HRS_ELC_ALL_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 44 *for bytes 1 to 78.*				
79	0	52224	DATA	DATA: Electron Counts 64 Energies x 51 Bins The 51 Bins are: 0 - 15 : E060 Look Directions 0-15 16 - 31 : E180 Look Directions 0-15 32 - 47 : E300 Look Directions 0-15 48 : E060 Background Anode 49 : E180 Background Anode 50 : E300 Background Anode (See SIS document for a figure.) [Note: E300 was turned off in 2016, so look directions 32-47 and 50 are usually populated with the MISSING_CONSTANT value of 65535.]
6607	0	8	MAG_TIME_DIR	MAG Time Direction. Indicates whether the MAG_TIME_LATENCY object indicates a past or future latency. 0: (time_received - time_in_packet) 1: (time_in_packet - time_received)
6608	0	8	MAG_LOOK_DIR [In earlier drafts was known as MAG_ANODE.]	MAG Look direction (0 to 47), e.g. which electron bin the MAG vector fell on. E060 has bins 0 to 15 E180 has bins 16 to 31 E300 has bins 32 to 47 (See DATA object for description of bins and look directions, and see the SIS document for a figure.) Note: Background anodes (per sensor) are not included for this mapping.
6609	0	16	MAG_FIRST_DFL	MAG First Deflection. First Deflection value written to the electron sweep table for the sensor the mag vector landed on. (1-bit range/gain (MSb) + 12-bit DAC value).
6611	0	8	MAG_ELEVATION	MAG elevation. Elevation angle of the mag vector, rounded to the nearest degree.

Byte	Bit	Length (bits)	Name	Description
6612	0	8	MAG_TIME_LATENCY	MAG Time Latency. Time difference (in 250 ms blocks) between the timestamp of the MAG vector and the time it was received by JADE. Values map to: 0 = 0.000 - 0.249 s 1 = 0.250 - 0.499 s 2 = 0.500 - 0.749 s 3 = 0.750 - 0.999 s etc. 14 = 3.500 - 3.749 s 15 = 3.750 - infinity s
6613	0	32	MAG_TIMESTAMP_WHOLE	MAG_TIMESTAMP_WHOLE Timestamp (whole second) for MAG vector. MAG timestamp subsecond is not returned in JADE high rate science electron packets. Referenced from 12:00UTC 2000/01/01. [May be affected by a Juno Time Stutter.]
6617	0	96	MAG_VECTOR	MAG vector in nT, 3 components [X, Y, Z] MAG range is +/- 16 G, hence limits. The coordinate system is spacecraft based, with: +X is between E060 and E300, along the 0 degree mark where E060 anode 0 starts +Y is 90 degrees, between E060 anodes 11 and 12, +Z is the spin axis. Note: these are signed integers. A Mag vector of [0 0 0] has four meanings: [Meanings 1 and 2 require MAG_TIMESTAMP_WHOLE = 0] 1) JADE never received a mag vector at all. (So initialized to 0s.) 2) A 25s timeout has expired without JADE receiving a MAG vector over a threshold magnitude. [Meanings 3 and 4 require MAG_TIMESTAMP_WHOLE > 0] 3) The threshold parameter was set to 0 nT. (Some early HVCO1 check-out data may have this.) 4) The broadcast message was corrupted and the magnitude and components mismatched.

6.2.9.1.3 JAD_L20_HVE_ELC_ALL_*

The electron product for high voltage engineering is PACKETID 0x1E and includes data from all three electron sensors.
The DATA object is 1-D, 51 anodes (over 1 energy), and is described in Table 49.
[The one energy is really 64 energy steps where all 64-steps are at the same fixed energy.]

*Table 49: Format of Level 2 data records for JAD_L20_HVE_ELC_ALL_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 44 *for bytes 1 to 78.*				
79	0	1632	DATA	DATA: Electron Counts 1 Energy x 51 Bins The 51 Bins are: 0 - 15 : E060 Look Direction 0-15 16 - 31 : E180 Look Directions0-15 32 - 47 : E300 Look Directions0-15 48 : E060 Background Anode 49 : E180 Background Anode 50 : E300 Background Anode (See SIS document for a figure.) 16-bit counter at 1 energy step over ACCUMULATION_TIME. Theoretical range (with max Accumulation of 1800s) is 0 to 7549632000, greater than a 4-byte unsigned int. It is extremely unlikely to fill 4-bytes, but if it does it will simply roll over to zero and keep going.

6.2.9.1.4 JAD_L20_LRS_ELC_ANY_*

The electron products for low rate science (PACKETID 0x68, 0x6A and 0x6B) cover all three electron sensors in the same file but one record contains data from one electron sensor only. At any time only one sensor will have a record.

This product is not about anodes, but electron Spin Phase sectors (different to ion Spin Phase sectors). For electron spin sectoring, there is a many-many relationship between anodes and spin sectors; an anode will travel through many sectors, and many anodes will contribute to each sector.

The formula for mapping sensor look directions into electron spin-phase sectors is:

spin_sector = ((*spin_phase* + *sensor_id* + 7.5 * *look_direction_id)* / 7.5 - 8) MOD 48

where:

spin_sector (electron) is 0 to 47 (rounded down to an integer).

spin_phase is in the range 0 to 360 degrees (angle from last crossing) (S/C reports in SHK files the angle to next crossing, a decreasing number, hence this spin-phase is 360 degrees minus that.)

sensor_id is either 60, 180, or 300, depending on which sensor it is for.

look_direction_id is 0 to 15 (also known as *sensor_loop_index* in operations documents) and is the look direction of anode *anode_id*, where *look_direction_id = 15- anode_id*.

anode_id (or *a_id*) is one of the 16 anodes of the given sensor, 0-15.

7.5 degrees is the width of one anode.

(15 – *anode_id*) is to correct for the look directions of the anodes.

Simplifying the equation (the last one is the one listed in the PDS file description):

=> *spin_sector* = ((*spin_phase* + *sensor_id* + 7.5 * (15 - *anode_id))*) / 7.5 - 8) MOD 48

=> *spin_sector* = ((*spin_phase* + *sensor_id*) / 7.5 + (15 - *anode_id)* - 8) MOD 48

=> *spin_sector* = ((*spin_phase* + *sensor_id*) / 7.5 + 7 - *anode_id*) MOD 48

Note that during FSW4.0 (April 2015 data only) the flight software had reversed anode mapping which messed up this calculation (see ISSUES object) (this reverse mapping affected all earlier FSW versions too, which are not covered in this PDS volume). If using FSW4.00/April 2015 data for this product (cruise solar wind only, no Jupiter science use) do not trust the spin sector calculation (in Level 3 data *DIM2_AZIMUTH_DESPUN* is replaced with fill values for FSW4.00). The only science use is to sum over spin phase sector to reduce the array to energy by time only. This was fixed in FSW4.10 (uploaded prior to the August 2015 data), from when this calculation was done correctly.

The DATA object is 2-D, 64 energies x 48 Electron Spin Phase sectors (from one electron sensor), and is described in

*Table 50: Format of Level 2 data records for JAD_L20_LRS_ELC_ANY_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	98304	DATA	DATA: Counts 64 Energies x 48 Electron Spin Phase Sectors. The formula for mapping anodes into spin-phase sectors is described in full in the PDS JADE SIS and simplifies to: SP_sector = ((s_phase + s_id)/7.5 + 7 - a_id) MOD 48 where: SP_sector (electron spin phase sector) is 0 to 47 (rounded down to an integer). s_phase is spin phase, 0 to 360 degrees. s_id is the sensor in question, either 60, 180, or 300. a_id is one of the 16 anodes of the given sensor, 0-15. (This is anode, not the look direction.) 7.5 degrees is the width of one anode. Note the data units are rates (counts per views), are floats rather than integers, and are fractions of 1/512. Note 2: Rate is independent of accumulation time. Note 3: If the data is from FSW 4.00 (April 2015 only, when anodes were reversed - see ISSUES object) then the SP_sector calculation was done incorrectly. If you must use this FSW4.00 data, sum over electron spin phase sector to reduce the data to energy by time and use that.
12359	0	32	MAG_TIMESTAMP_WHOLE	MAG_TIMESTAMP_WHOLE Whole-second timestamp of last received MAG vector *before* data collection start. Referenced from 12:00UTC 2000/01/01.
12363	0	16	MAG_TIMESTAMP_SUB	MAG_TIMESTAMP_SUB Sub-second timestamp of last received MAG vector *before* data collection start. A value of 65535 could be real or a MISSING_CONSTANT, however it is MISSING_CONSTANT only if MAG_TIMESTAMP_WHOLE = 0, e.g. WHOLE and SUB must both be real or both be MISSING_CONSTANT. Unit: Microseconds scaled to 16 bits.

Byte	Bit	Length (bits)	Name	Description
12365	0	8	MAG_COUNT_VALID	MAG_COUNT_VALID Count of valid (above threshold and not saturated) MAG vectors between start of *previous* packet and start of this packet. Note: This saturates at 255. e.g if there is a 600s accumulation period, and the MAG vector is given every 2-seconds, then that's 300 counts. If all are valid then that 300 will be expressed as 255, however MAG_COUNT_INVALID would still be zero.
12366	0	8	MAG_COUNT_INVALID	MAG_COUNT_INVALID Count of invalid (below threshold or saturated) MAG vectors between start of *previous* packet and start of this packet. Note: This saturates at 255. e.g if there is a 600s accumulation period, and the MAG vector is given every 2-seconds, then that's 300 counts. If all are invalid then that 300 will be expressed as 255, however MAG_COUNT_VALID would still be zero.
12367	0	96	MAG_VECTOR	Last received MAG vector in nT before data collection start: 3 components [X, Y, Z] MAG range is +/- 16 G, hence limits. The coordinate system is spacecraft based, with: +X is between E060 and E300, along the 0 degree mark where E060 anode 0 starts +Y is 90 degrees, between E060 anodes 11 and 12, +Z is the spin axis. Note: these are signed integers. A Mag vector of [0 0 0] has four meanings: [Meanings 1 and 2 require MAG_TIMESTAMP_WHOLE = 0. The MAG_COUNT_VALID and MAG_COUNT_INVALID objects can help distinguish meaning 1 from 2.] 1) JADE never received a mag vector at all. (So initialized to 0s.) e.g. MAG_COUNT_VALID = 0 for this record. 2) A 25s timeout has expired without JADE receiving a MAG vector over a threshold magnitude. [Meanings 3 and 4 require MAG_TIMESTAMP_WHOLE > 0 3) The threshold parameter was set to 0 nT. (Some early HVCO1 check-out data may have this.) 4) The broadcast message was corrupted and the magnitude and components mismatched.

Byte	Bit	Length (bits)	Name	Description
12379	0	32	BACKGROUND _COUNTS	Background counts (NOT a background rate). The background counter for this record's electron sensor (see ESENSOR object to know which sensor). This is a total count, not a rate. This is a 16-bit counter over 64 energies over the accumulation time (up to 1800 seconds), which means it could roll over the 4-byte word. i.e. 4294967296 = 0 However this is unlikely, and even if so, should be obvious from the visible background in object DATA.
12383	0	16	ESENSOR	ESENSOR - which one of the three electron sensors is this record for. Values can only be 60, 180 or 300 for electron sensor E060, E180 or E300 respectively. Note: each sensor also has a different PACKETID.

6.2.9.2 Ion Species Data

The JADE ion sensor has 12 anodes, as shown in the following figure.

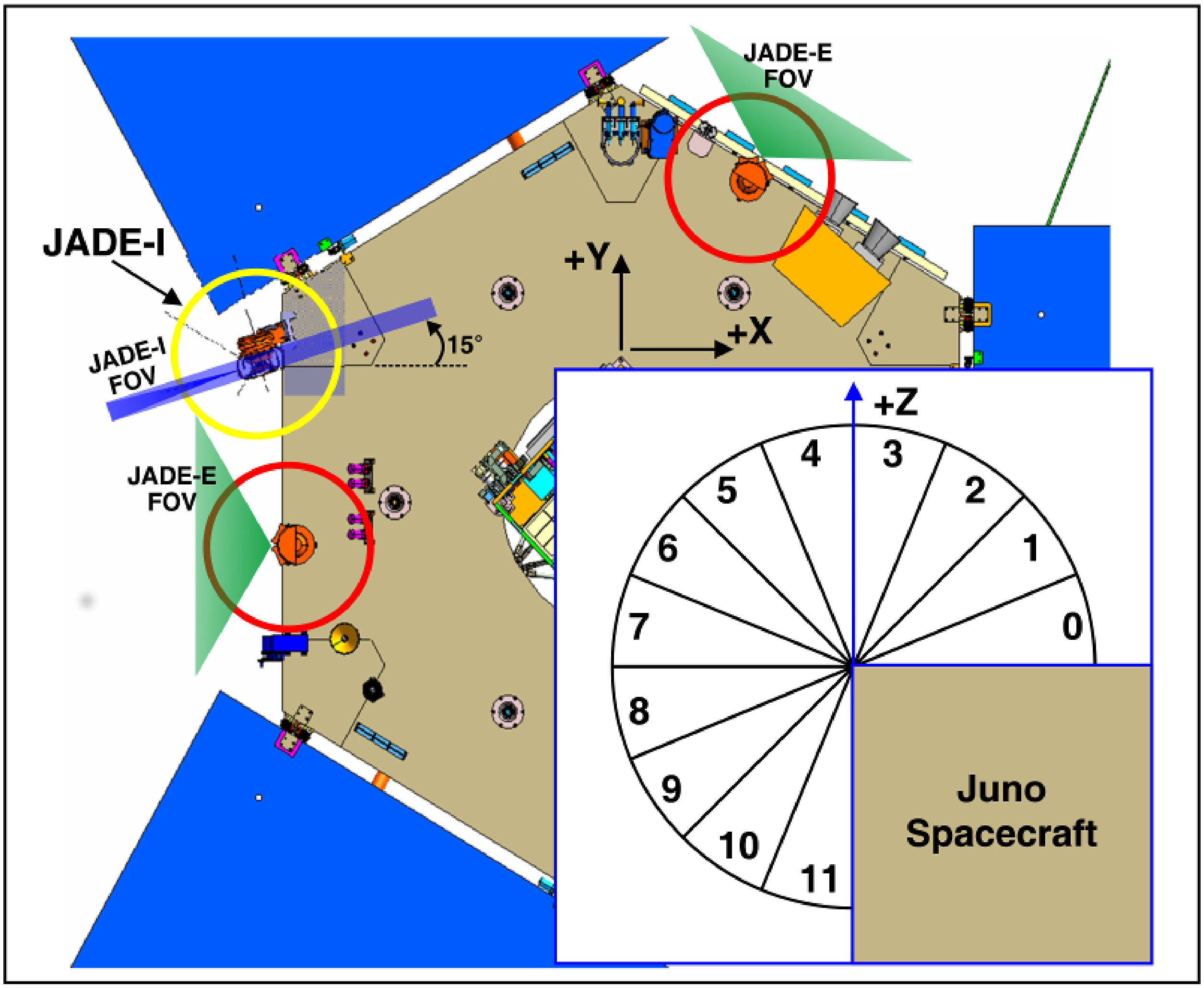


Figure 9: JADE-I Anodes.

6.2.9.2.1 JAD_L20_HRS_ION_ANY_*

The ion species products for high rate science cover PACKETIDs 0x80-0x87. Each ion species has its own packet; therefore several packets of different species may have the same time stamp. The DATA object is 2-D, 32 energies x 12 anodes, and is described in Table 51.

*Table 51: Format of Level 2 data records for JAD_L20_HRS_ION_ANY_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	6144	DATA	DATA: Counts 32 Energies x 12 Anodes

6.2.9.2.2 JAD_L20_HVE_ION_ALL_*

The ion species product for high voltage engineering is PACKETID 0x18.
The DATA object is 2-D, 8 ion species x 12 anodes (over 1 energy), and is described in Table 52. [The one energy is really 32 energy steps where all 32-steps are at the same fixed energy.]

*Table 52: Format of Level 2 data records for JAD_L20_HVE_ION_ALL_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	3072	DATA	DATA: Counts 8 Species x 12 anodes (x1 Energy) Species is 0-7, ion anodes 0-11. (16-bit counter, summed over 32 identical energy steps over accumulation period.)

6.2.9.2.3 JAD_L20_LRS_ION_ANY_* and JAD_L20_CAL_ION_ANY_*

The ion species products for low rate science (PACKETID 0x60-0x67) and calibration modes (PACKETID 0x70-0x77) are identical. Each ion species has its own packet; therefore several packets of different species may have the same time stamp.
The DATA object is 2-D, 32 energies x 78 ion spin phase sectors (note that ion spin phase sectors in FSW4 are different to those described in the original JADE instrument paper that was for FSW3). Ion spin phase sectors [0-77] are a factor of ion anode and the spin phase at the start of the record. The ion spin phase sector mapping is shown in Figure 10 and also within Table 53 in the description of the DATA object (where a range of 0-12 means 0 <= range < 12); it is different to the electron spin phase sectors.

Spin phase of Juno's +x axis is the angle **from** the last ECLIPJ2000 +z ('north') crossing, which increases over time and is shown in light red on the bottom of Figure 10. Juno reports in JADE operations SHK files the angle to next crossing, a decreasing number; hence this spin phase is 360 degrees minus that. The actual spin phase of JADE-I is shown in the dark red on the top of Figure 10, and is always 195 degrees greater than the equivalent spin phase of Juno's +x axis.

Spin phase of JADE-I instrument (Degrees)

<table>
<tr><td></td><td>195</td><td>207</td><td>219</td><td>231</td><td>243</td><td>255</td><td>267</td><td>279</td><td>291</td><td>303</td><td>315</td><td>327</td><td>339</td><td>351</td><td>3</td><td>15</td><td>27</td><td>39</td><td>51</td><td>63</td><td>75</td><td>87</td><td>99</td><td>111</td><td>123</td><td>135</td><td>147</td><td>159</td><td>171</td><td>183</td></tr>
<tr><td>Anode 0</td><td colspan="30" rowspan="4">Anodes 0-3 not reported as part of LRS Species map</td></tr>
<tr><td>Anode 1</td></tr>
<tr><td>Anode 2</td></tr>
<tr><td>Anode 3</td></tr>
<tr><td>Anode 4</td><td colspan="10">0</td><td colspan="10">1</td><td colspan="10">2</td></tr>
<tr><td>Anode 5</td><td colspan="6">3</td><td colspan="4">4</td><td colspan="6">5</td><td colspan="4">6</td><td colspan="6">7</td><td colspan="4">8</td></tr>
<tr><td>Anode 6</td><td colspan="2">9</td><td colspan="2">10</td><td colspan="2">11</td><td colspan="2">12</td><td colspan="2">13</td><td colspan="2">14</td><td colspan="2">15</td><td colspan="2">16</td><td colspan="2">17</td><td colspan="2">18</td><td colspan="2">19</td><td colspan="2">20</td><td colspan="2">21</td><td colspan="2">22</td><td colspan="2">23</td></tr>
<tr><td>Anode 7</td><td colspan="2">24</td><td colspan="2">25</td><td colspan="2">26</td><td colspan="2">27</td><td colspan="2">28</td><td colspan="2">29</td><td colspan="2">30</td><td colspan="2">31</td><td colspan="2">32</td><td colspan="2">33</td><td colspan="2">34</td><td colspan="2">35</td><td colspan="2">36</td><td colspan="2">37</td><td colspan="2">38</td></tr>
<tr><td>Anode 8</td><td colspan="2">39</td><td colspan="2">40</td><td colspan="2">41</td><td colspan="2">42</td><td colspan="2">43</td><td colspan="2">44</td><td colspan="2">45</td><td colspan="2">46</td><td colspan="2">47</td><td colspan="2">48</td><td colspan="2">49</td><td colspan="2">50</td><td colspan="2">51</td><td colspan="2">52</td><td colspan="2">53</td></tr>
<tr><td>Anode 9</td><td colspan="2">54</td><td colspan="2">55</td><td colspan="2">56</td><td colspan="2">57</td><td colspan="2">58</td><td colspan="2">59</td><td colspan="2">60</td><td colspan="2">61</td><td colspan="2">62</td><td colspan="2">63</td><td colspan="2">64</td><td colspan="2">65</td><td colspan="2">66</td><td colspan="2">67</td><td colspan="2">68</td></tr>
<tr><td>Anode 10</td><td colspan="4">69</td><td colspan="6">70</td><td colspan="4">71</td><td colspan="6">72</td><td colspan="4">73</td><td colspan="6">74</td></tr>
<tr><td>Anode 11</td><td colspan="10">75</td><td colspan="10">76</td><td colspan="10">77</td></tr>
<tr><td></td><td>0</td><td>12</td><td>24</td><td>36</td><td>48</td><td>60</td><td>72</td><td>84</td><td>96</td><td>108</td><td>120</td><td>132</td><td>144</td><td>156</td><td>168</td><td>180</td><td>192</td><td>204</td><td>216</td><td>228</td><td>240</td><td>252</td><td>264</td><td>276</td><td>288</td><td>300</td><td>312</td><td>324</td><td>336</td><td>348</td></tr>
</table>

Spin phase of Juno +X axis (Degrees)

Figure 10: The Ion Spin Phase Sector mapping to ion anode and spin phase (start angles).

Be careful with this product when ACCUMULATION_TIME = 30 for a record. There are the 78 ion spin phase sectors, however for a 30 second accumulation it is possible that not all 78 spin phase sectors will be observed. This is because Juno's spin period may be 29.2 to 30.8 seconds, and at time of writing has always been over 30 seconds. If the spin period was exactly 30.0 seconds, then all spin-phase sectors for anodes 6 to 9 would be observed once in a 30 second accumulation. If the spin period is less than 30 seconds, it's possible for 4 or 8 sectors to be skipped. E.g. if spin phase sector 9 is skipped, then so is spin phase sector 24, 39 and 54 as they are all at the same spin phase. Or if the spin period is greater than 30 seconds and the spin started on exactly zero spin-phase, then the accumulation period may simply end before spin phase sector 23 (and 38, 53, 68 as all the same spin phase) is reached. Identifying which spin phase sectors may be missed or unobserved is tricky as it depends on both the spin period and exact spin-phase at the start of the accumulation period, neither of which is known for a level 2 product.

This means that 4 or 8 spin-phase sectors may not be observed in a record with an ACCUMULATION_TIME of 30 seconds, and is usually just 4. The assumption is that LRS or CAL ion species data covers the full sky of 4π steradians, however this is not true when some spin-phase sectors are missing. If the accumulation time is greater than 30 (that is 60, 120, 150, 300 or 600) seconds then all spin phase sectors are viewed at least once during the accumulation, therefore none are unobserved.

If a spin phase sector is unobserved then a value of zero counts/view is returned. Unfortunately, this is indistinguishable from an observed spin-phase sector that simply did not measure any ions. Hence there is potential confusion when presented with zero counts/view (or counts per second for Level 3 files), was this spin phase sector measured but counted no ions, or was it not measured at all.

If it is known that some spin phase sectors are missing, one could use the values from the equivalent spin phase sectors on the next record. However, the LRS and CAL ion species data has many measured zero counts/view, making it impossible to identify which are unobserved when in the solar wind or in a low count region. In high count regions where there are naturally some counts (even if they are background counts) in most spin phase sectors then it is easy to pick out the 4 or 8 elements of a record that are zero because there are no other zeros. In such case, if 4 zeros, they should be in spin phase sector numbers that are 15 apart, e.g. the same spin phase like spin phase sectors 14, 29, 44 and 59. If 8 zeros, then there should be two sets of numbers 15 apart. Also, for a given time stamp there will be multiple records with different ion species numbers, e.g. PACKET_SPECIES of 3, 4 and 5. For the same time stamp, all species should be missing the same spin phase sector numbers.

*Table 53: Format of Level 2 data records for JAD_L20_LRS_ION_ANY_* and JAD_L20_CAL_ION_ANY_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	79872	DATA	DATA: Counts per view 32 Energies x 78 Ion Spin Phase Sectors The formula for mapping anodes into spin-phase sectors is described in the PDS JADE SIS and as follows: Each Spin Phase Sector has contributions from multiple spin phases, but always the same anode. The spin phase is calculated from the start of the record. Only anodes 4-11 are used, 0-3 are not reported. There are 78 Spin Phase sectors [0-77] over the eight anodes and thirty 12-degree wide sectors, with spin phase sectors given in the following table of anode by start spin phase:

Ion Start Spin Phase (Degrees)	Ion Anode 4	5	6	7	8	9	10	11
195-207	0	3	9	24	39	54	69	75
207-219	0	3	9	24	39	54	69	75
219-231	0	3	10	25	40	55	69	75
231-243	0	3	10	25	40	55	69	75
243-255	0	3	11	26	41	56	70	75
255-267	0	3	11	26	41	56	70	75
267-279	0	4	12	27	42	57	70	75
279-291	0	4	12	27	42	57	70	75
291-303	0	4	13	28	43	58	70	75
303-315	0	4	13	28	43	58	70	75
315-327	1	5	14	29	44	59	71	76
327-339	1	5	14	29	44	59	71	76
339-351	1	5	15	30	45	60	71	76
351-003	1	5	15	30	45	60	71	76
003-015	1	5	16	31	46	61	72	76
015-027	1	5	16	31	46	61	72	76
027-039	1	6	17	32	47	62	72	76
039-051	1	6	17	32	47	62	72	76
051-063	1	6	18	33	48	63	72	76
063-075	1	6	18	33	48	63	72	76
075-087	2	7	19	34	49	64	73	77
087-099	2	7	19	34	49	64	73	77
099-111	2	7	20	35	50	65	73	77
111-123	2	7	20	35	50	65	73	77
123-135	2	7	21	36	51	66	74	77
135-147	2	7	21	36	51	66	74	77
147-159	2	8	22	37	52	67	74	77
159-171	2	8	22	37	52	67	74	77
171-183	2	8	23	38	53	68	74	77
183-195	2	8	23	38	53	68	74	77

[The onboard software triggers on the spin phase of the s/c +X axis, but JADE-I is 195 degrees further around, so the Ion Start Spin Phase starts at 195.]

The meaning of each species is described in the JADE instrument paper.
Note the data units are rates (counts per views), are floats rather than integers, and are fractions of 1/512.
Note 2: Rate is independent of accumulation time.

6.2.9.3 Ion Time of Flight Data

6.2.9.3.1 JAD_L20_HLC_ION_TOF_*

The ion time of flight products for high rate science, low rate science and calibration mode are all of the same format and in the same file; covering PACKETID 0x69, 0x79 and 0x89. [Technically, the high rate science data product is of 2-byte values, whereas low rate/calibration data are 4-byte values, however the high rate data is up-cast to 4-bytes to make them identical.] The DATA object is 2-D, 32 energies x 96 TOF channels, and is described in Table 54.

*Table 54: Format of Level 2 data records for JAD_L20_HLC_ION_TOF_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	98304	DATA	DATA: Time of Flight Counts 32 Energies x 96 TOF channels. The last 3 channels have special meanings. The 96 TOF channels are created from the Medium Resolution set of 128 channels, binned as follows: [0]: Medium Resolution TOF channels 0 - 1 [1]: Medium Resolution TOF channel 2 [2]: Medium Resolution TOF channel 3 [3]: Medium Resolution TOF channel 4 ... [59]: Medium Resolution TOF channel 60 [60]: Medium Resolution TOF channel 61 [61]: Medium Resolution TOF channels 62 - 63 [62]: Medium Resolution TOF channels 64 - 65 ... [70]: Medium Resolution TOF channels 80 - 81 [71]: Medium Resolution TOF channels 82 - 83 [72]: Medium Resolution TOF channel 84 [73]: Medium Resolution TOF channel 85 [74]: Medium Resolution TOF channel 86 [75]: Medium Resolution TOF channel 87 [76]: Medium Resolution TOF channel 88 [77]: Medium Resolution TOF channels 89 - 90 [78]: Medium Resolution TOF channels 91 - 92 ... [89]: Medium Resolution TOF channels 113 - 114 [90]: Medium Resolution TOF channels 115 - 116 [91]: Medium Resolution TOF channels 117 - 118 [92]: Medium Resolution TOF channels 119 - 124 [93]: TOF with Start overload. (Medium Resolution TOF channel 125) [94]: TOF value below minimum resolution. (Medium Resolution TOF channel 126) [95]: TOF too long. (Medium Resolution TOF channel 127)

6.2.9.3.2 JAD_L20_HVE_ION_TOF_*

The ion time of flight product for high voltage engineering is PACKETID 0x19.
The DATA object is 1-D, 128 TOF channels (over 1 energy), and is described in Table 55.
[The one energy is really 32 energy steps where all 32-steps are at the same fixed energy.]

*Table 55: Format of Level 2 data records for JAD_L20_HVE_ION_TOF_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	4096	DATA	DATA: Time of Flight Counts 128 TOF channels (x 1 Energy) The last 3 channels have special meanings. The 128 TOF channels are counters are: [0]: Medium Resolution TOF channel 0 [1]: Medium Resolution TOF channel 1 … [123]: Medium Resolution TOF channel 123 [124]: Medium Resolution TOF channel 124 [125]: TOF with Start overload. (Medium Resolution TOF channel 125) [126]: TOF value below minimum resolution. (Medium Resolution TOF channel 126) [127]: TOF too long. (Medium Resolution TOF channel 127) (16-bit counter, summed over 32 identical energy steps over accumulation period.)

6.2.9.4 Ion Logicals Data

6.2.9.4.1 JAD_L20_HLC_ION_LOG_*

The ion logicals products for high rate science, low rate science and calibration mode are all of the same format and in the same file; covering PACKETID 0x6C, 0x7C and 0x8C. [Technically, the high rate science data product is of 2-byte values, whereas low rate/calibration data are 4-byte values, however the high rate data is upcast to 4-bytes to make them identical.] The DATA object is 2-D, 32 energies x 25 logicals, and is described in Table 56, which runs over 2 pages despite only being for the DATA object.

The 25 logical counters here are the same for JAD_L20_HVE_ION_LOG_*.

*Table 56: Format of Level 2 data records for JAD_L20_HLC_ION_LOG_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	25600	DATA	DATA: Counts 32 Energies x 25 Logical counters. The 25 Logical counters are: [0]: Anode 0 Events [1]: Anode 1 Events [2]: Anode 2 Events … [10]: Anode 10 Events [11]: Anode 11 Events [12]: Background Events The above 13 logicals are raw count hits, independent of whether a TOF Event has begun. Adjacent and Non-Adjacent hits will be counted in both anodes. As such, anode counts can exceed All Stops [15] counts. The Background anode [12] is not included in Adjacent and Non-Adjacent calculations. [13]: Start Overload Start signal exceeds threshold level. [14]: All Starts Independent of whether a TOF Event has begun, usually starts a TOF Event. [15]: All Stops Independent of whether a TOF Event has begun, usually ends a TOF Event. If an event is seen on multiple anodes this counter is still only incremented once, therefore this is usually less than the sum of anodes 0 to 11. The Background anode is not included in All Stops, just anodes 0 to 11. [16]: Non-Adjacent Anodes This is either two non-neighbor anodes (anodes 0-11 only), or more than 2 anodes. *Continues on next page.*

Byte	Bit	Length (bits)	Name	Description
				Continues from previous page. [17]: Adjacent Anodes A count hit was measured in neighboring anodes; other products (e.g. Ion Species) will assign this to just the lower anode. [18]: Stop without Start A stop signal was received before a TOF Event was initiated by a start. [19]: Dual Start A TOF Event had started but one or more other start signals were received before a stop signal or the TOF Event overflowed. [20]: Start in Process Time The number of TOF Events started, can be less than All Starts [14]. [21]: TOF Underflow Received a stop event before 1 tap, that is 1.6ns, the base unit of TOF times. [22]: TOF Overflow No stop signal arrived within timeout of 330ns. [23]: Invalid TOF Event If the TOF Event is measured in 1 anode (anodes 0-11 only) or two neighboring anodes (anodes 0-11 only) it is valid. Otherwise it is invalid, unless it was an underflow in which case the Underflow [21] counter is increased instead of this counter (i.e. an Underflow event is considered valid). Therefore, if the event is not an Underflow event, it will be invalid if one of these three situations is met: - hit in more than two anodes, or - hit in two non-neighbor anodes, or - no anodes hit at all. The latter is different to overflow events [22] which are considered valid. The Background anode is not considered in any of these calculations. [24]: Event Strobe The number of TOF Events completed, by a stop signal or over/underflow, usually the same as Start in Process Time [20]. For the above, a TOF Event is a start signal followed by either a stop signal or timeout. Note: This file can contain data from Low Rate Science, MCP Calibration or High Rate Science modes. High Rate Science have a maximum counts of 65535 per accumulation unlike the others with a maximum of 117963000.

6.2.9.4.2 JAD_L20_HVE_ION_LOG_*

The ion logicals product for high voltage engineering is PACKETID 0x1C.
The DATA object is 1-D, 25 logicals (over 1 energy), and is described in Table 57.
[The one energy is really 32 energy steps where all 32-steps are at the same fixed energy.]

The 25 logical counters here are the same for JAD_L20_HLC_ION_LOG_*.

*Table 57: Format of Level 2 data records for JAD_L20_HVE_ION_LOG_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	800	DATA	DATA: Counts 1 Energy x 25 Logical counters. The 25 Logical counters are: [0]: Anode 0 [1]: Anode 1 [2]: Anode 2 [3]: Anode 3 [4]: Anode 4 [5]: Anode 5 [6]: Anode 6 [7]: Anode 7 [8]: Anode 8 [9]: Anode 9 [10]: Anode 10 [11]: Anode 11 [12]: Background [13]: Start Overload [14]: All Starts [15]: All Stops [16]: Non-Adjacent Anodes [17]: Adjacent Anodes [18]: Stop without Start [19]: Dual Start [20]: Start in Process Time [21]: TOF Underflow [22]: TOF Overflow [23]: Invalid TOF Event [24]: Event Strobe (16-bit counter, summed over 32 identical energy steps over accumulation period.) See the DESCRIPTION of DATA in JAD_L20_HLC_ION_LOG files for a better description of the 25 logicals.

6.2.9.5 Ion Direct Events Data

The ion direct events products for high rate science, low rate science, calibration mode and high voltage engineering are all of the same format and in the same file; covering PACKETID 0x1F, 0x6F, 0x7F and 0x8F.

Direct Event (DE) data records the full resolution data on an event-by-event basis, as opposed to the other data products which are collapsed based on product type over an accumulation period. Whereas JAD_L20_HLC_ION_TOF_* data will collect the number of incident ions at each energy step falling within each TOF bin over an accumulation period, Direct Event data record the anode, TOF bin, and ESA step on an event by event basis. Because of this the data volume of Direct Events is too large to provide a continuous record and only a subset is returned. Direct Events will be of most use in performing spot validation of the other data products.

6.2.9.5.1 JAD_L20_ALL_ION_DER_*

The DE-Words contained in the JAD_L20_ALL_ION_DER_* data require decoding, and have been decoded in the JAD_L20_ALL_ION_DES_* file, see Table 60, which we expect science users to use in preference.

*Table 58: Format of Level 2 data records for JAD_L20_ALL_ION_DER_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	16	DE_COL_SUB_SEQ_COUNT	Direct Events Collection sub-sequence count. Resets to 0 at the start of the playback of a new collection cycle. Increments for each produced packet, before data policing, thus acting as an indicator for data policing loss. Maximum value is 19,999 unless the data is taken from High Rate Science (HRS), when the maximum is 3599. See PACKET_MODE or PACKETID objects to see if in HRS.
73	0	34656	DATA	DATA: Direct Event Two-Byte Words Array of 16-bit raw direct events. A Direct Event is information about each specific particle that hit the Ion sensor. [See PDS JADE SIS document for JAD_L20_ALL_ION_DES for a table on how to decode this 2-byte word.] Note, not all 2166 bytes are used, see DE_SIZE Object, and this object has to be padded to size.
4405	0	16	DE_SIZE	Array size of Direct Events to use. The above data array can vary in size, but PDS records must be a fixed size, so the end is padded with fill values. This value tells you how many values should be used (starting from the beginning).

6.2.9.5.2 JAD_L20_ALL_ION_DES_*

This is the same data as for the JAD_L20_ALL_ION_DER_* products (from the same JADE packet IDs), except the DATA object's data is split out in to its many meanings. Table 59 describes how the two-byte word can either be an *event* word, a *boundary* word, a *sweep marker* word or a *fill value* (occasionally required for padding the DER DATA to a fixed size), and then how to split up the bit pattern for each. Each JAD_L20_ALL_ION_DER_* DATA word then becomes an entire JAD_L20_ALL_ION_DES_* record. As such, one JAD_L20_ALL_ION_DER_* record can become (up to) 2,166 JAD_L20_ALL_ION_DES_* records. If the DER DATA word was fill then no JAD_L20_ALL_ION_DES record is written. Technically the Sweep Number is a 14-bit long value, however it has a limit of 1800, which results in bits 13 to 11 always being zero. Direct Events will be of most use in performing spot validation of the other data products.

Since FSW 4.00 the ion deflectors are turned off during HRS, such that 'DFL Step' is always zero, hence the 'Boundary Word' is now essentially identical format for all telemetry modes. This following table is still valid, and is also true for earlier FSW versions (e.g. FSW 3).

Table 59: Description of DATA two-byte words for JAD_L20_ALL_ION_DER_* files to show how it is split out for the JAD_L20_ALL_ION_DES_* files.

Bit number	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Event Word	0	TOF								Anode ID				QF2	0	QF0
Boundary Word LRS/CAL/HVE	1	0	0	0	0	0	0	0	ESA Step					0	0	0
Boundary Word HRS	1	0	0	0	0	0	0	0	ESA Step					DFL Step		
Sweep Marker Word	1	1	Sweep Number (max 1800) 0	 0	 0	 Sweep Number										
Fill Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The format of the JAD_L20_ALL_ION_DES_* data records is given on the next page, Table 60, and extends over 2 pages.

*Table 60: Format of Level 2 data records for JAD_L20_ALL_ION_DES_**

Byte	Bit	Length (bits)	Name	Description
See Level 2 binary header from Table 42 *and* Table 43 *for bytes 1 to 70.*				
71	0	16	DE_COL_SUB_SEQ_COUNT	Direct Events Collection sub-sequence count. Resets to 0 at the start of the playback of a new collection cycle. Increments for each produced packet, before data policing, thus acting as an indicator for data policing loss. Maximum value is 19,999 unless the data is taken from High Rate Science (HRS), when the maximum is 3599. See PACKET_MODE or PACKETID objects to see if in HRS.
73	0	8	DE_BAD	Bad Direct Event Two-Byte Word = 0 is good, Direct Event word is valid. = 1 is bad, Direct Event word is invalid. The two-byte DE Words found in JAD_L20_ALL_ION_DER_* files are split out bit by bit in JAD_L20_ALL_ION_DES_* files, however only certain combinations are valid. If a non-valid bit combination is found, all objects for that record in this file are set to their MISSING_CONSTANT value. [Possible reason for invalid words are SEUs (single event upsets) where a bit is altered in memory. While very rare, we believe such a SEU occurred in Direct Event data at SCLK 494493538 (2015-245). SEUs may occur in other JADE products too, but the bit structure of direct events make their identification easier.]
74	0	8	DE_EVENT0_BOUNDARY1_MARKER2	Direct Event, or Boundary, or Sweep Marker, Word. 0 = Direct Event Word 1 = Boundary Word 2 = Sweep Marker Word 255 = Fill Value - all other DE_* objects should also be their MISSING_CONSTANT VALUE To decode the original bit pattern of a DE_WORD: 0 if bit 15 = 0 1 if bit 15 = 1 and bit 14 = 0 2 if bit 15 = 1 and bit 14 = 1
75	0	16	DE_SWEEP_NUMBER	Direct Event Sweep Number. (bits 13-0) Value is only given when DE_EVENT0_BOUNDARY1_MARKER2 is 2, otherwise equals MISSING_CONSTANT.
77	0	8	DE_ESA_STEP	Direct Event ESA Step. (bits 7-3) Value is only given when DE_EVENT0_BOUNDARY1_MARKER2 is 1, otherwise equals MISSING_CONSTANT.
78	0	8	DE_DFL_STEP	Direct Event DFL Step. (bits 2-0) Value is only given when DE_EVENT0_BOUNDARY1_MARKER2 is 1, otherwise equals MISSING_CONSTANT.

Byte	Bit	Length (bits)	Name	Description
79	0	16	DE_TOF	Direct Event TOF value. (bits 14-7) 0-252: Valid TOF measurement (min_TOF to 330ns). 253 : TOF with Start overload. 254 : TOF value below minimum resolution. 255 : TOF too long. 65535: Fill value = MISSING_CONSTANT (Value in telemetry is 1 byte (0-255) only, but upcast to 2 bytes here to allow a MISSING_CONSTANT value to be added.) Value is only given when DE_EVENT0_BOUNDARY1_MARKER2 is 0, otherwise equals MISSING_CONSTANT.
81	0	8	DE_ANODE_ID	Direct Event Anode ID. (bits 6-3) 0-11: Valid Anode ID, 0-11. 12-13: Reserved (should never be seen). 14 : No Anode ID between Start and Stop. 15 : Two non-adjacent Anodes IDs between Start and Stop. Value is only given when DE_EVENT0_BOUNDARY1_MARKER2 is 0, otherwise equals MISSING_CONSTANT.
82	0	8	DE_QUALITY_FLAG_2	Direct Event Quality Flag 2: (bit 2) Two Adjacent Anode IDs between Start and Stop. 0 = Flag not triggered, 1 = Flag triggered. Value is only given when DE_EVENT0_BOUNDARY1_MARKER2 is 0, otherwise equals MISSING_CONSTANT.
83	0	8	DE_QUALITY_FLAG_1	Direct Event Quality Flag 1: (bit 1) Reserved - should be 0. Value is only given when DE_EVENT0_BOUNDARY1_MARKER2 is 0, otherwise equals MISSING_CONSTANT.
84	0	8	DE_QUALITY_FLAG_0	Direct Event Quality Flag 0: (bit 0) Additional Start(s) between Start and Stop. 0 = Flag not triggered, 1 = Flag triggered. Value is only given when DE_EVENT0_BOUNDARY1_MARKER2 is 0, otherwise equals MISSING_CONSTANT.

6.2.10 Level 3 data files for file versions 01, 02 and 03

This section (6.2.10) and sub-sections are only for Level 3 file versions 01, 02 and 03.
If you are after Level 3 file version 04, go to section 6.2.11.

The Level 3 data files are binary and have files ending in the extension .DAT. Accompanying them in the same directory are the label files with the same filename but the extension .LBL.

For example, the PDS file pairs will have the following paths in the Volume:

ROOT/DATA/*yyyy*/*yyyyddd*/*subdir*/JAD_L30_*aaa*_*bbb*_*ccc*_*uuu*_*yyyyddd*_V*nn*.DAT
ROOT/DATA/*yyyy*/*yyyyddd*/*subdir*/JAD_L30_*aaa*_*bbb*_*ccc*_*uuu*_*yyyyddd*_V*nn*.LBL

The format file (same filename minus the date part, but including the version number, with the extension .FMT) accompanying (and already listed within) the LBL files are usually found in the LABEL directory at the root of the volume – however it was decided to exclude this LABEL directory (and therefore exclude FMT files) as they are redundant and may be copy/pasted out of the LBL files. [FMT files are made locally for JADE file production, but do not get to the PDS.]

See section 3.1 for the explanation of JAD_L30_*aaa*_*bbb*_*ccc*_*uuu*_*yyyyddd*_V*nn*, and *subdir* is the subdirectory name given in Table 17.

There are currently 7 different Level 3 product types, see Table 61 for their sizes, but they are similar and they all have the same objects (per version) as a header. File versions 01, 02 and 03 are all the same format. File version 04 (and future versions) are a different format, and are discussed later in section 6.2.11.

Table 61: Size of a record of each Level 3 product, by version number[a,b]

Version (*nn*)	Product	Bytes per record	Objects per record
01, 02, 03	JAD_L30_HLS_ION_LOG_CNT_V*nn*	83488	47
01, 02, 03	JAD_L30_HLS_ION_TOF_CNT_V*nn*	100476	56
01, 02, 03	JAD_L30_HRS_ELC_ALL_CNT_V*nn*	160042	48
01, 02, 03	JAD_L30_HRS_ELC_TWO_CNT_V*nn*	106790	48
01, 02, 03	JAD_L30_HRS_ION_ANY_CNT_V*nn*	40224	47
01, 02, 03	JAD_L30_LRS_ELC_ANY_CNT_V*nn*	160036	49
01, 02, 03	JAD_L30_LRS_ION_ANY_CNT_V*nn*	259872	47
Version (*nn*)	**Product**	**Bytes per record**	**Objects per record**
04	JAD_L30_HLS_ION_LOG_CNT_V*nn*	45120	51
04	JAD_L30_HLS_ION_TOF_CNT_V*nn*	98228	58
04	JAD_L30_HRS_ELC_ALL_CNT_V*nn*	86346	52
04	JAD_L30_HRS_ELC_TWO_CNT_V*nn*	57670	52
04	JAD_L30_HRS_ION_ANY_CNT_V*nn*	21824	51
04	JAD_L30_LRS_ELC_ANY_CNT_V*nn*	86340	53
04	JAD_L30_LRS_ION_ANY_CNT_V*nn*	140096	51

(a) Level 3 versions 01 to 03 are an identical format.
(b) Versions not listed are expected to have the version 04 format.

To save space in this document, Table 63 gives the 34-object header for the binary files for Level 3 products versions 01, 02 and 03, which is then used throughout. This is the same for all, except the PACKETID (which can change within a product type for Level 3 data) that gives a different description for each packet, shown in blue, and the last 4 objects that have the same names but different sizes. The rest of the data product is the same format (floats) but may have different sizes. The UTC entries are not side by side due to PDS rules requiring multi-byte words to start on even byte boundaries, so are spaced by 1-byte words.

Efforts were made to keep the objects as similar as possible (both in name and dimensions), as shown in Figure 11. Some may consider this redundant but this is deliberately done so that the same code may be used on different datasets. For example a 64 by 48 object may only contain 64 unique values that change with the 1st dimension during low rate science files, however during high rate science files both the 1st and 2nd dimension values change – since these objects are the same dimension the same code may then be used to analyze both high and low rate science files.

In order to have fewer products than level 2 had, like ones were grouped together to give just 7 products per unit, with the unit of counts per second being the base file, that files with other units are to be created from. Data from high voltage engineering and calibration modes are excluded from level 3 data, as they are not designed for science use (possibly with highly variable MCPs voltages for MCP tests).

Level 3 data should be scientifically useful data, however there is still an object called ISSUES. This is for occasions where the data is scientifically valid, but may not be similar to its neighbors. For instance, the data may be accumulating records over 30 second accumulation times, but the last record was during a mode change so there's only 13 seconds. The data for those 13 seconds are valid, but for consistency the end user may wish to disregard and only use the full 30 second data that's available. This ISSUES object allows such occurrences to be flagged easily.

If a level 2 high rate or low rates science record is unsuitable for science work, a level 3 record may still be created, however the DATA object will be replaced with MISSING_CONSTANT fill values. This is to allow a user to know that high or low rate data was deliberately excluded, but does exist in level 2 data. However when calibration mode data is excluded (as not for science), no equivalent record of fill values will exist in the level 3 data.

The MISSING_CONSTANT for the objects DATA, DATA_SIGMA, BACKGROUND and BACKGROUND_SIGMA is -1 (not -999999) in Level 3 versions 01, 02 and 03 data (but not version 04). See section 0 for more details.

Table 62 lists the Level 3 products and which Level 2 products were used to get them. There are no high voltage engineering data in level 3 (no JAD_L20_HVE*), nor ion direct events (no JAD_*aaa*_ION_DER nor JAD_*aaa*_ION_DES).

Table 62: Mapping Level 2 data files to Level 3 data files

Level 2 Data Product	Path	Level 3 Data Product
JAD_L20_HRS_ELC_ALL	----------->	JAD_L30_HRS_ELC_ALL_CNT JAD_L30_HRS_ELC_TWO_CNT
JAD_L20_LRS_ELC_ANY	----------->	JAD_L30_LRS_ELC_ANY_CNT
JAD_L20_HLC_ION_LOG	Remove CAL data	JAD_L30_HLS_ION_LOG_CNT
JAD_L20_HRS_ION_ANY	----------->	JAD_L30_HRS_ION_ANY_CNT
JAD_L20_LRS_ION_ANY	----------->	JAD_L30_LRS_ION_ANY_CNT
JAD_L20_HLC_ION_TOF	Remove CAL data	JAD_L30_HLS_ION_TOF_CNT

As ion species records go in the same level 3 products, it is possible to have consecutive records with the same time stamp. The difference will be in the PACKETID that tells you which particular ion species that record is for. Likewise JAD_L30_LRS_ELC_ANY_CNT may contain records from any of the 3 electron sensors, however a given time will only ever have a record from one sensor record.

Note that the LBL/FMT files describe DATA, DATA_SIGMA, BACKGROUND, BACKGROUND_SIGMA, DIM1_*, DIM2_* and transformation matrices DESPUN_SC_TO_J2000 and J2000_TO_RTP as 2D or 3D containers (containers in containers that hold a scalar). If you read the object in as a 1D vector then it should be reformed by the user to a 2D or 3D array. The 1D ordering is based on c, in that the last dimension changes fastest, i.e. if a 1D array is x=[1, 2, 3, 4, 5, 6] and that should be a 3x2 array y, then:

y[0][0] = 1; y[0][1] = 2; y[1][0] = 3; y[1][1] = 4; y[2][0] = 5; y[2][1] = 6;

Object	Data Type	Total Number of Bytes	JAD_L30_HRS_ELC_ALL	JAD_L30_HRS_ELC_TWO	JAD_L30_LRS_ELC_ANY	JAD_L30_CAL_ELC_ALL	JAD_L30_HRS_ION_ANY	JAD_L30_LRS_ION_ANY	JAD_L30_CAL_ION_ANY	JAD_L30_HLS_ION_TOF	JAD_L30_CAL_ION_TOF	JAD_L30_HLS_ION_LOG	JAD_L30_CAL_ION_LOG
DIM0_UTC	char[21]	21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PACKETID	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM0_UTC_UPPER	char[21]	21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PACKET_MODE	int8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM0_UTC_LOWER	char[21]	21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PACKET_SPECIES	int8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ACCUMULATION_TIME	uint16[1]	2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DATA_UNITS	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOURCE_BACKGROUND	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOURCE_DEAD_TIME	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOURCE_MAG	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOURCE_JADE_METAKERNEL	int16[1]	2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOURCE_JADE_CALIB	int16[1]	2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FSW_VERSION	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_R	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_R_UPPER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_R_LOWER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LAT	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LAT_UPPER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LAT_LOWER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LOCAL_TIME	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LOCAL_TIME_UPPER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LOCAL_TIME_LOWER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_JUPITER_J2000XYZ	float[3]	12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_VEL_JUPITER_J2000XYZ	float[3]	12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_VEL_ANGULAR_J2000XYZ	float[3]	12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_SPIN_PERIOD	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DESPUN_SC_TO_J2000	float[3,3]	36	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
J2000_TO_JSSXYZ	float[3,3]	36	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
J2000_TO_JSSRTP	float[3,3]	36	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MCP_VOLTAGE	float	4, 8 or 12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ISSUES	uint32	4 or 8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TIMESTAMP_WHOLE	uint32	4 or 8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TIMESTAMP_SUB	uint16	2 or 4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DATA	float[64,n]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DATA_SIGMA	float[64,n]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BACKGROUND	float[64,n]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BACKGROUND_SIGMA	float[64,n]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM1_E	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM1_E_UPPER	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM1_E_LOWER	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM2_ELEVATION	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM2_ELEVATION_UPPER	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM2_ELEVATION_LOWER	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM2_AZIMUTH_DESPUN	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM2_AZIMUTH_DESPUN_UPPER	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM2_AZIMUTH_DESPUN_LOWER	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM3_TOF	float[n=93]	372								✓	✓		
DIM3_TOF_UPPER	float[n=93]	372								✓	✓		
DIM3_TOF_LOWER	float[n=93]	372								✓	✓		
TOF_WITH_START_OVERLOAD	float[64]	256								✓	✓		
TOF_WITH_START_OVERLOAD_SIGMA	float[64]	256								✓	✓		
TOF_TOO_SHORT	float[64]	256								✓	✓		
TOF_TOO_SHORT_SIGMA	float[64]	256								✓	✓		
TOF_TOO_LONG	float[64]	256								✓	✓		
TOF_TOO_LONG_SIGMA	float[64]	256								✓	✓		
MAG_VECTOR	float[3]	12	✓	✓	✓	✓							
ESENSOR	uint16[1]	2			✓								
Number of Objects			48	48	49	48	47	47	47	56	56	47	47

Figure 11: Breaking out the JADE Level 3 Version 01, 02 and 03 products in to the different PDS Objects to allow similarities to be drawn.
Grey columns represent calibration files for JADE operations use that will not go to the PDS. Blue text values do not need to be in level 3 files, but aids cross comparison with level 2 data, and red text are extra values that may be useful. m = n for all but TOF products, where m = 1 because of the 3rd TOF dimension.

The following table (over 7 pages) describes the header that is identical for all the following data version 01, 02 and 03 products (and is based on Level 3 Version 01 FMT files). The names and word type (int/float/etc.) for all level 3 version 01, 02 and 03 data is also summarized in Figure 11. Any text in *red italics* is a note that is not in the FMT file, while any text in **blue boldface** may change depending on the product (usually just the product ID or species number). This color system will apply for format tables throughout the rest of section 6.2.

Table 63: Format of Level 3 data record header for Versions 01, 02 and 03

Byte	Length (bytes)	Name	Fmt*	Units	Description
1	21	DIM0_UTC	UTC string	Time	UTC timestamp at center (not start) of record. Format is yyyy-dddTHH:MM:SS.sss where yyyy = year, ddd = day of year, HH = hour, MM = minute, SS.sss = decimal seconds to millisecond resolution. Note: Duration of record can be found in S.I. seconds by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse this with the ACCUMULATION_TIME object, which is the number of spacecraft clock ticks for accumulation. While 1 tick is approximately 1 second, it is not identical.
22	1	PACKETID	uint8	None	Packet ID (DPID), Data Product Identifier **Low Rate Science – Electron** **One electron sensor per record:** **Sensor E060 is PACKETID = 104 (0x68)** **Sensor E180 is PACKETID = 106 (0x6A)** **Sensor E300 is PACKETID = 107 (0x6B)** **[There is no PACKETID = 105]** *Note: A value of 255 indicates Unknown, which can be used for higher order products that use a mix of packets.*
23	21	DIM0_UTC_UPPER	UTC string	Time	0th Dimension of DATA: Time - upper limit. See DIM0_UTC for description.

Byte	Length (bytes)	Name	Fmt*	Units	Description
44	1	PACKET_MODE	int8	None	Packet Mode, describes type of data telemetry. -2 = HSK / Housekeeping Engineering (Level 2 only) -1 = HVE / High Voltage Engineering (Level 2 only) 0 = CAL / MCP Calibration Science (Level 2 only) 1 = LRS / Low Rate Science 2 = HRS / High Rate Science 3 = DRS / DeRived Science from LRS and/or HRS 127 = Unknown 254 = Wrong - but HSK, see below. (Level 2 only) 255 = Wrong - but HVE, see below. (Level 2 only) (Note, this could also be calculated via PACKETID.) If you have 254 or 255 then your code is incorrect, check you read a signed byte, rather than unsigned.
45	21	DIM0_UTC_LOWER	UTC string	Time	0th Dimension of DATA: Time - lower limit. See DIM0_UTC for description.
66	1	PACKET_SPECIES	int8	None	Packet Species, describes type of plasma data. -1 = electrons 0 = ion species 0, SP0 1 = ion species 1, SP1 2 = ion species 2, SP2 3 = ion species 3, SP3 4 = ion species 4, SP4 5 = ion species 5, SP5 6 = ion species 6, SP6 7 = ion species 7, SP7 8 = Sum of SP3, SP4 and SP5 9 = All ions /* or any ion, e.g. TOF and LOG */ 10 = Single ion species derived from TOF data 127 = Unknown 255 = Wrong - but electrons, see below. If you have 255 then your code is incorrect, check you read a signed byte, rather than unsigned.

Byte	Length (bytes)	Name	Fmt*	Units	Description
67	2	ACCUMULATION_TIME	uint16	SCLK ticks	Accumulation Time. Number of seconds over which the data in this product was collected (Science Program). *Note: Duration of record can be found in S.I. seconds by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse this with the ACCUMULATION_TIME object, which is the number of spacecraft clock ticks for accumulation.* *While 1 tick is approximately 1 second, it is not identical.* *ACCUMULATION_TIME is left in spacecraft clock ticks to both aid matching with the level 2 data and to help filtering for data taken in a particular mode.*
69	1	DATA_UNITS	uint8	None	Data units correspond to: 0 = All counts in the accumulation period 1 = All counts divided by number of views /* 0 and 1 are for Level 2 data only – but keeping the numbering convention */ 2 = Counts per second /* S.I. science units: */ 3 = Differential Energy Flux [1/(m2 sr s)] 4 = Differential Number Flux [1/(m2 sr s J)] 5 = Phase Space Density [m-6 s3] /* Convenient (non-S.I.) science units: */ 6 = Differential Energy Flux [1/(cm2 sr s)] 7 = Differential Number Flux [1/(cm2 sr s keV)] 8 = Phase Space Density [cm-6 s3] /* As new products are developed this list will increase */ /* If a number is not listed, */ /* try a LBL/FMT file from a recent date. */ 255 = Unknown.
70	1	SOURCE_BACKGROUND	uint8	None	Source of Background values (see BACKGROUND object) that have been removed from the DATA object. 0 = None: No background has been removed 1 = Background anode (electron sensors only) /* As new background removal methods are developed this list will increase */ 255 = Unknown.

Byte	Length (bytes)	Name	Fmt*	Units	Description
71	1	SOURCE_DEAD_TIME	uint8	None	Source of Dead Time Correction Method 0 = None: Data has not been Dead Time corrected. 255 = Unknown.
72	1	SOURCE_MAG	uint8	None	Source of MAG data Except case 0 and 1, PAYLOAD (pl) co-ordinate MAG files were used at 1s (or 2s if no 1s) resolution. 0 = None: No MAG data in this product. 1 = From Juno JADE's Level 2 files. (From spacecraft and therefore uncalibrated.) This is independent to JADE Level 2 version number as it does not change with versions. [Note MAG data in JADE files may be affected by the Juno time stutter.] 3n = Juno's MAG's Level 3 version n calibrated files, e.g. 34 means version 4, so: 30 = From Juno MAG's Level 3 version 00 quicklook payload files. (These are temporary files not in PDS.) 31 = From Juno MAG's Level 3 version 01 calibrated payload files. 32 = From Juno MAG's Level 3 version 02 calibrated payload files. Likewise 33 to 39 being Level 3 version 3 to 9. 255 = Unknown. If you see a number not listed above, there may be later versions of MAG data - find the latest available LBL file for this product and see what that has listed.

Byte	Length (bytes)	Name	Fmt*	Units	Description
73	2	SOURCE_JADE_METAKERNEL	int16	None	The JADE SPICE metakernel used to get the time, position, velocity, orientation and transformation objects in this file. The metakernel lists the many individual spice kernels used, which are archived by NAIF and not in this PDS volume. The JADE SPICE metakernel may be found in the CALIB directory of this PDS volumne, with filenames of: JAD_L30_SPICE_METAKERNEL_*nnnnn*.TXT where *nnnnn* is the SOURCE_JADE_METAKERNEL object number (with leading zeros and positive). If any of the kernels within the metakernel are not reconstucted (but reference or predicted) for the time in question, this value will be negative. Within the JADE PDS archive this value should always be positive.
75	2	SOURCE_JADE_CALIB	int16	None	The JADE calibration files list used to convert the engineering units of Level 2 data to the scientific units in this file. Similar to the SPICE metakernel list, this lists the many individual calibration files used, each of which may be found in the CALIB directory on this PDS volumne. This list may be found in the CALIB directory of this PDS volumne, with filenames of: JAD_L30_CALIB_LIST_nnnnn.TXT where nnnnn is the SOURCE_JADE_CALIB object number (with leading zeros and positive). If any of the calibration files listed are not final at the time in question, this value will be negative. (Newer calibration files will have a higher version and simply be listed in a newer SOURCE_JADE_CALIB file.) Within the JADE PDS archive this value should always be positive. However a version 00 file (for team use or uploaded to JSOC, not PDS) may have negative values with predicted positions/orientations/transformations.
77	4	FSW_VERSION	f	None	Flight Software version used. Number should be to 2 decimal places, with rounding. e.g. 4.00, 4.10, 4.20.

Byte	Length (bytes)	Name	Fmt*	Units	Description
81	4	SC_POS_R	f	R_J	Juno radial distance from Jupiter. (1 Rj = 71492.0 km) [Values may be greater than VALID_MAXIMUM during cruise to Jupiter before primary mission.]
85	4	SC_POS_R_UPPER	f	R_J	Juno radial distance from Jupiter - upper limit. See SC_POS_R for description.
89	4	SC_POS_R_LOWER	f	R_J	Juno radial distance from Jupiter - lower limit. See SC_POS_R for description.
93	4	SC_POS_LAT	f	Degrees	Juno Latitude above Jupiter. (0 = Equatorial)
97	4	SC_POS_LAT_UPPER	f	Degrees	Juno Latitude above Jupiter - upper limit. See SC_POS_LAT for description.
101	4	SC_POS_LAT_LOWER	f	Degrees	Juno Latitude above Jupiter - lower limit. See SC_POS_LAT for description.
105	4	SC_POS_LOCAL_TIME	f	Hours	Juno Local Time from Jupiter. 00 = Midnight 06 = Dawn 12 = Noon 18 = Dusk
109	4	SC_POS_LOCAL_TIME_UPPER	f	Hours	Juno Local Time from Jupiter - upper limit. See SC_POS_LOCAL_TIME for description.
113	4	SC_POS_LOCAL_TIME_LOWER	f	Hours	Juno Local Time from Jupiter - lower limit. See SC_POS_LOCAL_TIME for description.
117	12	SC_POS_JUPITER_J2000XYZ	f	km	Juno position from Jupiter in J2000 cartesian co-ordinates [x,y,z] (units km). [Values may be outside of VALID_MIN/MAX range (~140Rj) during cruise to Jupiter before primary mission.]
129	12	SC_VEL_JUPITER_J2000XYZ	f	km/s	Juno Velocity with respect to Jupiter in J2000 Cartesian co-ordinates [Vx,Vy,Vz] (units km/s).
141	12	SC_VEL_ANGULAR_J2000XYZ	f	rads/s	Juno Angular Velocity in cartesian co-ordinates [AVx,AVy,AVz] (units radians/s). (This is calculated with the SPICE ckgpav command where ref=J2000. SPICE defines it as 'This is the axis about which the reference frame tied to the instrument is rotating in the right-handed sense.')
153	4	SC_SPIN_PERIOD	f	Seconds	Juno spin period (seconds). This is not useful during spacecraft maneuvers.

Byte	Length (bytes)	Name	Fmt*	Units	Description
157	36	DESPUN_SC_TO_J2000	f	None	Rotation matrix from despun spacecraft co-ordinates to J2000. This is a 3x3 matrix, but if read in as a 1x9 stream then the 1D stream is [a,b,c, d,e,f, g,h,i] and the 2D matrix would be [a,b,c d,e,f g,h,i]
193	36	J2000_TO_JSS XYZ	f	None	Rotation matrix from J2000 co-ordinates to JSS xyz (JSS = Jupiter-De-Spun-Sun, see SIS for details). This is a 3x3 matrix, but if read in as a 1x9 stream then the 1D stream is [a,b,c, d,e,f, g,h,i] and the 2D matrix would be [a,b,c d,e,f g,h,i]
229	36	J2000_TO_JSS RTP	f	None	Rotation matrix from J2000 co-ordinates to JSS RTP, where RTP is Jupiter centered right handed R-Theta-Phi. (JSS = Jupiter-De-Spun-Sun, see SIS for details.) This is a 3x3 matrix, but if read in as a 1x9 stream then the 1D stream is [a,b,c, d,e,f, g,h,i] and the 2D matrix would be [a,b,c d,e,f g,h,i]
265		MCP_VOLTAGE			The last 4 objects of this header all start at byte 265 and have the same names, but three different sizes depending on the JADE product. For the ion products go to Table 64. For the HRS electrons (all) go to Table 65, or Table 66 for HRS electrons (two). For the LRS electrons go to Table 67.
		ISSUES			
		TIMESTAMP_WHOLE			
		TIMESTAMP_SUB			

Fmt* is shortened for the table and is decoded in PDS format as: f = PC_REAL (float), uint8/uint16/uint32 are = one/two/four-byte LSB_UNSIGNED_INTEGER and int8/int16/int32 = one/two/four byte LSB_INTEGER.

Table 64: Format of Level 3 data record subheader for Level 3 ion products. for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
265	4	MCP_VOLTAGE	f	Volts	MCP Voltage on sensor.
269	8	ISSUES	uint32	None	Issues or potential issues in this data record. *[Two values for ions as this is the ISSUES object from both the ping and pong level 2 packets used to create this record.]...* The rest is a direct copy of the Level 2 ISSUES object, see Table 45 for description.
277	8	TIMESTAMP_WHOLE	uint32	Ticks	Timestamps (Whole Second) of JADE Level 2 packets used to make this Level 3 record. (Both the ping and pong level 2 packets.)
285	4	TIMESTAMP_SUB	uint16	Subticks	Timestamps (Subsecond) of JADE Level 2 packets used to make this Level 3 record. (Both the ping and pong level 2 packets.)

Table 65: Format of Level 3 data record subheader for JAD_L30_HRS_ELC_ALL_* for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
265	12	MCP_VOLTAGE	f	Volts	MCP Voltages on the three electron sensors, E060, E180 and E300 respectively.
277	4	ISSUES	uint32	None	Issues or potential issues in this data record. … The rest is a direct copy of the Level 2 ISSUES object, see Table 45 for description.
281	4	TIMESTAMP_WHOLE	uint32	Ticks	Timestamp (Whole Second) of JADE Level 2 packet used to make this Level 3 record.
285	2	TIMESTAMP_SUB	uint16	Subticks	Timestamp (Subsecond) of JADE Level 2 packet used to make this Level 3 record.

Table 66: Format of Level 3 data record subheader for JAD_L30_HRS_ELC_TWO_* for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
265	8	MCP_VOLTAGE	f	Volts	MCP Voltages on the two electron sensors in this product, E060 and E180 respectively.
273	4	ISSUES	uint32	None	Issues or potential issues in this data record. … The rest is a direct copy of the Level 2 ISSUES object, see Table 45 for description.
277	4	TIMESTAMP_WHOLE	uint32	Ticks	Timestamp (Whole Second) of JADE Level 2 packet used to make this Level 3 record.
281	2	TIMESTAMP_SUB	uint16	Subticks	Timestamp (Subsecond) of JADE Level 2 packet used to make this Level 3 record.

Table 67: Format of Level 3 data record subheader for JAD_L30_LRS_ELC_ANY_* for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
265	4	MCP_VOLTAGE	f	Volts	MCP Voltage on sensor.
269	4	ISSUES	uint32	None	Issues or potential issues in this data record. … The rest is a direct copy of the Level 2 ISSUES object, see Table 45 for description.
273	4	TIMESTAMP_WHOLE	uint32	Ticks	Timestamp (Whole Second) of JADE Level 2 packet used to make this Level 3 record.
277	2	TIMESTAMP_SUB	uint16	Subticks	Timestamp (Subsecond) of JADE Level 2 packet used to make this Level 3 record.

In general, the rest of the format for the different products have the same object names (see Figure 11), however their size (byte length) and start bytes will differ. The descriptions are also much the same when they have the same object name, with only DATA really changing (text that may alter between products is shown in **blue boldface**).

6.2.10.1 Electron Data for V01, V02 and V03

6.2.10.1.1 JAD_L30_HRS_ELC_ALL_CNT_* for V01, V02 and V03

The electron product for high rate science is PACKETID 0x8E and includes data from all three electron sensors.
The DATA object is 2-D, 64 energies x 48 look directions, and is described in Table 68, and continues over the next 4 pages.

This product is a combination of look directions from all 3 JADE-E sensors, but E300 was turned off in 2016, hence those anodes that would have been from E300 are populated with the MISSING_CONSTANT (-1) value.

Table 68: Format of Level 3 data records for JAD_L30_HRS_ELC_ALL_CNT for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 63 *and* Table 65 *for bytes 1 to 286.*					
287	12288	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 48 Look Directions. [Note: E300 was turned off in 2016, so the last 16 look directions (32-47) are usually populated with the MISSING_CONSTANT value of -1.]
12575	12288	DATA_SIGMA	f	Counts/s	DATA_SIGMA 1-sigma uncertainties on values in object DATA, such that true value = DATA +/- DATA_SIGMA. See DATA entry above for size information.
24863	12288	BACKGROUND	f	Counts/s	Background value removed from DATA. No further background removal is required. If you wish to do your own background removal, add this object to DATA then you can remove a background via your own method. The background values here were found from either a background anode or JADE's own ground method.
37151	12288	BACKGROUND_SIGMA	f	Counts/s	BACKGROUND_SIGMA 1-sigma uncertainties on values in object BACKGROUND, such that true value = BACKGROUND +/- BACKGROUND_SIGMA. See BACKGROUND entry above for size information.

Byte	Length (bytes)	Name	Fmt*	Units	Description
49439	12288	DIM1_E	f	eV/q	1st Dimension of DATA: Energy - center eV/q value. Upper and lower limits are given by the objects DIM1_E_UPPER and DIM1_E_LOWER.
61727	12288	DIM1_E_UPPER	f	eV/q	1st Dimension of DATA: Energy - upper eV/q limit. See DIM1_E for description.
74015	12288	DIM1_E_LOWER	f	eV/q	1st Dimension of DATA: Energy - lower eV/q limit. See DIM1_E for description.
86303	12288	DIM2_ELEVATION	f	Degrees	2nd Dimension of DATA: Spacecraft elevation - center value. Spacecraft elevation (degs) is analogous to latitude on a sphere. In spacecraft xyz co-ords: +z is equivalent to elevation = +90 degs -z is equivalent to elevation = -90 degs (The communication dish is directed along +z) xy-plane at z = 0 is equivalent to elevation = 0. Note, 2nd dimension is really look direction which has an elevation and azimuth; hence two objects describe this: DIM2_ELEVATION and DIM2_AZIMUTH_DESPUN.
98591	12288	DIM2_ELEVATION_UPPER	f	Degrees	2nd Dimension of DATA: S/C elevation - upper limit. See DIM2_ELEVATION for description.
110879	12288	DIM2_ELEVATION_LOWER	f	Degrees	2nd Dimension of DATA: S/C elevation - lower limit. See DIM2_ELEVATION for description.

Byte	Length (bytes)	Name	Fmt*	Units	Description
123167	12288	DIM2_AZIMUTH_DESPUN	f	Degrees	2nd Dimension of DATA: Despun S/C azimuth - center value. Spacecraft azimuth (degs) is analogous to longitude on a sphere. In spacecraft xyz co-ords: +x is equivalent to azimuth = 0 degs +y is equivalent to azimuth = 90 degs -x is equivalent to azimuth = 180 degs -y is equivalent to azimuth = 270 degs +x is equivalent to azimuth = 360 degs +y is equivalent to azimuth = 450 degs The 'Despun' azimuth angle varies because Juno spins, where azimuth = 0 is defined as +x when spin phase equals zero (e.g. despun x-z plane contains the ECLIPJ2000 north). The relationship between despun azimuth and spin phase (which decreases during a spin) is simply: Despun Azimuth = 360 degrees - Spin Phase Because a lower to upper limit could occur over a 360 degree boundary, the VALID_MINIMUM and VALID_MAXIMUM go from 0 to +720 degrees: e.g. [lower, center, upper] = [-10, 5, 20] would be given instead as = [350, 365, 380] Note, 2nd dimension is really look direction which has an elevation and azimuth; hence two objects describe this: DIM2_ELEVATION and DIM2_AZIMUTH_DESPUN.
135455	12288	DIM2_AZIMUTH_DESPUN_UPPER	f	Degrees	2nd Dimension of DATA: Despun S/C azimuth - upper limit. See DIM2_AZIMUTH_DESPUN for description.
147743	12288	DIM2_AZIMUTH_DESPUN_LOWER	f	Degrees	2nd Dimension of DATA: Despun S/C azimuth - lower limit. See DIM2_AZIMUTH_DESPUN for description.

Byte	Length (bytes)	Name	Fmt*	Units	Description
160031	12	MAG_VECTOR	f	nT	MAG vector in nT, 3 components [X, Y, Z] MAG range is +/- 16 G, hence limits. This xyz coordinate system is despun spacecraft; see the definitions of DIM2_ELEVATION and DIM2_AZIMUTH: +X is when [azimuth, elevation] = [0, 0] degrees, +Y is when [azimuth, elevation] = [90, 0] degrees, +Z is when elevation = 90 degrees.

6.2.10.1.2 JAD_L30_HRS_ELC_TWO_CNT_* for V01, V02 and V03

This is a repeat of the JAD_L30_HRS_ELC_ALL_CNT_* file, but with E300 data removed to provide a smaller (but still large) file, thus only contains E060 and E180 data. This product was introduced when it was decided not to use sensor E300 in flight operations, however the HRS electron data packet would still return zeros for E300.

The DATA object is 2-D, 64 energies x 32 look directions (rather than 48 look directions), and is described in Table 69.

Table 69: Format of Level 3 data records for JAD_L30_HRS_ELC_TWO_CNT for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 63 *and* Table 65 *for bytes 1 to 286.*					
283	8192	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 32 Look Directions.
8475	8192	DATA_SIGMA	f	Counts/s	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
16667	8192	BACKGROUND	f	Counts/s	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
24859	8192	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
33051	8192	DIM1_E	f	eV/q	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
41243	8192	DIM1_E_UPPER	f	eV/q	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
49435	8192	DIM1_E_LOWER	f	eV/q	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
57627	8192	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
65819	8192	DIM2_ELEVATION_UPPER	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
74011	8192	DIM2_ELEVATION_LOWER	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
82203	8192	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
90395	8192	DIM2_AZIMUTH_DESPUN_UPPER	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
98587	8192	DIM2_AZIMUTH_DESPUN_LOWER	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
106779	12	MAG_VECTOR	f	nT	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*

6.2.10.1.3 JAD_L30_LRS_ELC_ANY_CNT_* for V01, V02 and V03

The electron products for low rate science are PACKETIDs 0x68, 0x6A and 0x6B, and includes data from one electron sensor per record (only one sensor is on at any given time).
The DATA object is 2-D, 64 energies x 48 look directions, and is described in Table 70.
Practically there are only two differences between this and the JAD_L30_HRS_ELC_ALL_CNT_* file:

1) The MCP_VOLTAGE object is a singular value here (for the one sensor) as opposed to 3 values for the HRS case (one for each of the sensors). This in turn makes the start byte of all following objects 8 bytes earlier in the LRS product compared to the HRS product. The description of MCP_VOLTAGE in the FMT file is slightly different to reflect this.
2) This product has an extra object at the end; called ESENSOR that states which of the three sensors is in use (60, 180 or 300). This does not exist in the HRS product as the data array always includes all three sensors.

So the only difference between tables Table 68 and Table 70 are the first column byte values are offset by 8 (as indicated in the first red row), and Table 70 has the ESENSOR product at the end.

If using FSW4.00 (April 2015 only) data for this product (cruise solar wind only, no Jupiter science use) all *DIM2_AZIMUTH_DESPUN* values are replaced with the fill value 65535 due to the reverse anode mapping bug (see section 6.2.9.1.4).

Table 70: Format of Level 3 data records for JAD_L30_LRS_ELC_ANY_CNT for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 63 *and* Table 67 *for bytes 1 to 278.*					
279	12288	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 48 Look Directions. *This is the same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
12567	12288	DATA_SIGMA	f	Counts/s	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
24855	12288	BACKGROUND	f	Counts/s	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
37143	12288	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
49431	12288	DIM1_E	f	eV/q	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
61719	12288	DIM1_E_UPPER	f	eV/q	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
74007	12288	DIM1_E_LOWER	f	eV/q	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
86295	12288	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
98583	12288	DIM2_ELEVATION_UPPER	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
110871	12288	DIM2_ELEVATION_LOWER	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
123159	12288	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
135447	12288	DIM2_AZIMUTH_DESPUN_UPPER	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
147735	12288	DIM2_AZIMUTH_DESPUN_LOWER	f	Degrees	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
160023	12	MAG_VECTOR	f	nT	*Same description as from* Table 68 *for JAD_L30_HRS_ELC_ALL_CNT.*
160035	2	ESENSOR	uint16	None	ESENSOR - which one of the three electron sensors is this record for. Values can only be 60, 180 or 300 for electron sensor E060, E180 or E300 respectively. Note: each sensor also has a different PACKETID. *This object is NOT in the product for JAD_L30_HRS_ELC_ALL_CNT.*

6.2.10.2 Ion Species Data for V01, V02 and V03

6.2.10.2.1 JAD_L30_HRS_ION_ANY_CNT_* for V01, V02 and V03

The ion species products for high rate science cover PACKETIDs 0x80-0x87. Each ion species has its own packet; therefore several packets of different species may have the same time stamp. The DATA object is 2-D, 64 energies x 12 look directions, and is described in Table 71, and continues over the next 3 pages.

Table 71: Format of Level 3 data records for JAD_L30_HRS_ION_ANY_CNT for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 63 *and* Table 64 *for bytes 1 to 288.*					
289	3072	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 12 Look Directions.
3361	3072	DATA_SIGMA	f	Counts/s	DATA_SIGMA 1-sigma uncertainties on values in object DATA, such that true value = DATA +/- DATA_SIGMA. See DATA entry above for size information.
6433	3072	BACKGROUND	f	Counts/s	Background value removed from DATA. No further background removal is required. If you wish to do your own background removal, add this object to DATA then you can remove a background via your own method. The background values here were found from either a background anode or JADE's own ground method.
9505	3072	BACKGROUND_SIGMA	f	Counts/s	BACKGROUND_SIGMA 1-sigma uncertainties on values in object BACKGROUND, such that true value = BACKGROUND +/- BACKGROUND_SIGMA. See BACKGROUND entry above for size information.
12577	3072	DIM1_E	f	eV/q	1st Dimension of DATA: Energy - center eV/q value. Upper and lower limits are given by the objects DIM1_E_UPPER and DIM1_E_LOWER.
15649	3072	DIM1_E_UPPER	f	eV/q	1st Dimension of DATA: Energy - upper eV/q limit. See DIM1_E for description.
18721	3072	DIM1_E_LOWER	f	eV/q	1st Dimension of DATA: Energy - lower eV/q limit. See DIM1_E for description.

Byte	Length (bytes)	Name	Fmt*	Units	Description
21793	3072	DIM2_ELEVATION	f	Degrees	2nd Dimension of DATA: Spacecraft elevation - center value. Spacecraft elevation (degs) is analogous to latitude on a sphere. In spacecraft xyz co-ords: +z is equivalent to elevation = +90 degs -z is equivalent to elevation = -90 degs (The communication dish is directed along +z) xy-plane at z = 0 is equivalent to elevation = 0. Note, 2nd dimension is really look direction which has an elevation and azimuth; hence two objects describe this: DIM2_ELEVATION and DIM2_AZIMUTH_DESPUN.
24865	3072	DIM2_ELEVATION_UPPER	f	Degrees	2nd Dimension of DATA: S/C elevation - upper limit. See DIM2_ELEVATION for description.
27937	3072	DIM2_ELEVATION_LOWER	f	Degrees	2nd Dimension of DATA: S/C elevation - lower limit. See DIM2_ELEVATION for description.

Byte	Length (bytes)	Name	Fmt*	Units	Description
31009	3072	DIM2_AZIMUTH_DESPUN	f	Degrees	2nd Dimension of DATA: Despun S/C azimuth - center value. Spacecraft azimuth (degs) is analogous to longitude on a sphere. In spacecraft xyz co-ords: +x is equivalent to azimuth = 0 degs +y is equivalent to azimuth = 90 degs -x is equivalent to azimuth = 180 degs -y is equivalent to azimuth = 270 degs +x is equivalent to azimuth = 360 degs +y is equivalent to azimuth = 450 degs The 'Despun' azimuth angle varies because Juno spins, where azimuth = 0 is defined as +x when spin phase equals zero (e.g. despun x-z plane contains the ECLIPJ2000 north). The relationship between despun azimuth and spin phase (which decreases during a spin) is simply: Despun Azimuth = 360 degrees - Spin Phase Because a lower to upper limit could occur over a 360 degree boundary, the VALID_MINIMUM and VALID_MAXIMUM go from 0 to +720 degrees: e.g. [lower, center, upper] = [-10, 5, 20] would be given instead as = [350, 365, 380] Note, 2nd dimension is really look direction which has an elevation and azimuth; hence two objects describe this: DIM2_ELEVATION and DIM2_AZIMUTH_DESPUN.
34081	3072	DIM2_AZIMUTH_DESPUN_UPPER	f	Degrees	2nd Dimension of DATA: Despun S/C azimuth – upper limit. See DIM2_AZIMUTH_DESPUN for description.
37153	3072	DIM2_AZIMUTH_DESPUN_LOWER	f	Degrees	2nd Dimension of DATA: Despun S/C azimuth – lower limit. See DIM2_AZIMUTH_DESPUN for description.

6.2.10.2.2 JAD_L30_LRS_ION_ANY_CNT_* for V01, V02 and V03

The ion species products for low rate science (PACKETID 0x60-0x67). Each ion species has its own packet; therefore several packets of different species may have the same time stamp.
The DATA object is 2-D, 64 energies x 78 look directions, and is described in Table 72.
The basic format of this file is identical to the HRS counterpart, except there are 78 look directions here instead of 12. As such the start byte and lengths change, but the object names and descriptions are the same (except for the description of the DATA object).

Table 72: Format of Level 3 data records for JAD_L30_LRS_ION_ANY_CNT for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 63 *and* Table 64 *for bytes 1 to 288.*					
289	19968	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 78 Look Directions.
20257	19968	DATA_SIGMA	f	Counts/s	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
40225	19968	BACKGROUND	f	Counts/s	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
60193	19968	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
80161	19968	DIM1_E	f	eV/q	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
100129	19968	DIM1_E_UPPER	f	eV/q	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
120097	19968	DIM1_E_LOWER	f	eV/q	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
140065	19968	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
160033	19968	DIM2_ELEVATION_UPPER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
180001	19968	DIM2_ELEVATION_LOWER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
199969	19968	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
219937	19968	DIM2_AZIMUTH_DESPUN_UPPER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
239905	19968	DIM2_AZIMUTH_DESPUN_LOWER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*

6.2.10.3 Ion Time of Flight Data for V01, V02 and V03

6.2.10.3.1 JAD_L30_HLS_ION_TOF_CNT_* for V01, V02 and V03

The ion time of flight products for high and low rate science, covering PACKETIDs 0x69 and 0x89.
The DATA object is 3-D, 64 energies x 1 look direction x 93 TOF channels, and is described in Table 73 (over 2 pages). This product usually has 96 TOF channels with the last 3 having special meanings, but for level 3 data the last 3 channels have been removed and given their own objects within this file.

This product is usually considered to be a 2 dimensional array of energy by TOF channel. However all other JADE data is Energy by look direction, so to keep things similar, this product is a 3 dimensional array of 64 energies by 1 look direction by 93 TOF channels. There is only 1 look direction, but given the ion instrument covers 270 degrees field of view in elevation over the 12 anodes, and this product sums all 12 anodes, this leads to some interesting azimuth and elevation numbers. The DIM2_AZIMUTH objects will use the respective azimuth of anodes 4-11 (anodes 0-3 azimuths would normally be 180 degrees from those). However DIM2_ELEVATION will range from -90 to +180 degrees (spanning 270 degrees) with a center value of +45 degrees. As such, elevation of +90 to +180 is being used to describe the contribution of anodes 3, 2, 1 and 0 that are technically covering elevations of +90 down to 0 degrees but with an azimuth 180 degrees different.

The object names (and descriptions, DATA description excepted) are identical to the other level 3 ion products, but with 6 TOF only objects on the end. (Text that may alter between products is shown in **blue boldface**, e.g. version number of files should match the version number of the DAT files.)

Table 73: Format of Level 3 data records for JAD_L30_HLS_ION_TOF_CNT for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 63 *and* Table 64 *for bytes 1 to 288.*					
289	23808	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 1 Look Direction x 93 Channels. These channels are expressed as a duration in seconds in object DIM3_TOF, and for more details see the TOF_CHANNEL_TO_SECONDS_HLC_V03.CSV file in the CALIB directory of this PDS archive. The Level 2 data had 96 channels, those last 3 are now objects TOF_WITH_START_OVERLOAD, TOF_TOO_SHORT and TOF_TOO_LONG respectively.
24097	23808	DATA_SIGMA	f	Counts/s	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
47905	23808	BACKGROUND	f	Counts/s	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
71713	23808	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
95521	256	DIM1_E	f	eV/q	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
95777	256	DIM1_E_UPPER	f	eV/q	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
96033	256	DIM1_E_LOWER	f	eV/q	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
96289	256	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
96545	256	DIM2_ELEVATION_UPPER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
96801	256	DIM2_ELEVATION_LOWER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
97057	256	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
97313	256	DIM2_AZIMUTH_DESPUN_UPPER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
97569	256	DIM2_AZIMUTH_DESPUN_LOWER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
97825	372	DIM3_TOF	f	Seconds	3rd Dimension of DATA: Time Of Flight - center value. (Seconds)

Byte	Length (bytes)	Name	Fmt*	Units	Description
98197	372	DIM3_TOF_UPPER	f	Seconds	3rd Dimension of DATA: Time Of Flight - upper limit. See DIM3_TOF for description.
98569	372	DIM3_TOF_LOWER	f	Seconds	3rd Dimension of DATA: Time Of Flight - lower limit. See DIM3_TOF for description.
98941	256	TOF_WITH_START_OVERLOAD	f	Counts/s	TOF with start overload: Counts/Second A signal pulse that is too strong (above a threshold) in the electronics. Multiple start-overloads that occur within a 330ns event window are counted each time in the Logicals Start Overload, but only once here.
99197	256	TOF_WITH_START_OVERLOAD_SIGMA	f	Counts/s	TOF with start overload uncertainty: Counts/Second 1-sigma uncertainties on values in object TOF_WITH_START_OVERLOAD such that true value = TOF_WITH_START_OVERLOAD +/- TOF_WITH_START_OVERLOAD_SIGMA. See TOF_WITH_START_OVERLOAD entry above for size information.
99453	256	TOF_TOO_SHORT	f	Counts/s	TOF too short: Counts/Second TOF underflow: Count of TOF measurements that did not timeout, but resulted in a measurement smaller than the sensor could measure.
99709	256	TOF_TOO_SHORT_SIGMA	f	Counts/s	TOF too short uncertainty: Counts/Second 1-sigma uncertainties on values in object TOF_TOO_SHORT such that true value = TOF_TOO_SHORT +/- TOF_TOO_SHORT_SIGMA. See TOF_TOO_SHORT entry above for size information.
99965	256	TOF_TOO_LONG	f	Counts/s	TOF too long: Counts/Second TOF overflow: Count of TOF measurements that resulted in no stop signal arriving within 330ns of the start signal.
100221	256	TOF_TOO_LONG_SIGMA	f	Counts/s	TOF too long uncertainty: Counts/Second 1-sigma uncertainties on values in object TOF_TOO_LONG such that true value = TOF_TOO_LONG +/- TOF_TOO_LONG_SIGMA. See TOF_TOO_LONG entry above for size information.

6.2.10.4 Ion Logicals Data for V01, V02 and V03

6.2.10.4.1 JAD_L30_HLS_ION_LOG_CNT_* for V01, V02 and V03

The ion logicals products for high and low rate science, covering PACKETID 0x6C and 0x8C. The DATA object is 2-D, 64 energies x 25 logicals (each with variable look directions), and is described in Table 74 (over 3 pages).

Given the ion instrument covers 270 degrees field of view in elevation, this leads to some interesting azimuth and elevation numbers, as elevation can range from -90 to +180 degrees; see the descriptions below. e.g. if Azimuth is 200 degrees and elevation is 100 degrees, that's equivalent to an azimuth of 20 (200-180) degrees and an elevation of 80 (180-100) degrees. That is anode 0 will have an azimuth 180 degrees from anode's 7, which is described in the DIM2 objects, however the logicals that combine all individual anodes the DIM2 values will use the azimuth from anodes 4-11 for all, but the elevations range will be -90 to +180 degrees.

The 25 logical counters here are the same as for level 2 data.

Table 74: Format of Level 3 data records for JAD_L30_HLS_ION_LOG_CNT for V01, V02 and V03

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 63 *and* Table 64 *for bytes 1 to 288.*					
289	6400	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 25 Logicals. The 25 Logical counters are: [0]: Anode 0 Events [1]: Anode 1 Events [2]: Anode 2 Events … [10]: Anode 10 Events [11]: Anode 11 Events [12]: Background Events The above 13 logicals are raw count hits, independent of whether a TOF Event has begun. Adjacent and Non-Adjacent hits will be counted in both anodes. As such, anode counts can exceed All Stops [15] counts. The Background anode [12] is not included in Adjacent and Non-Adjacent calculations. [13]: Start Overload Start signal exceeds threshold level. [14]: All Starts Independent of whether a TOF Event has begun, usually starts a TOF Event. [15]: All Stops Independent of whether a TOF Event has begun, usually ends a TOF Event. If an event is seen on multiple anodes this counter is still only incremented once, therefore this is usually less than the sum of anodes 0 to 11. The Background anode is not included in All Stops, just anodes 0 to 11. [16]: Non-Adjacent Anodes This is either two non-neighbor anodes (anodes 0-11 only), or more than 2 anodes. [17]: Adjacent Anodes A count hit was measured in neighboring anodes; other products (e.g. Ion Species) will assign this to just the lower anode. [18]: Stop without Start A stop signal was received before a TOF Event was initiated by a start. *Continues on next page.*

Byte	Length (bytes)	Name	Fmt*	Units	Description
					Continues from previous page. [19]: Dual Start A TOF Event had started but one or more other start signals were received before a stop signal or the TOF Event overflowed. [20]: Start in Process Time The number of TOF Events started, can be less than All Starts [14]. [21]: TOF Underflow Received a stop event before 1 tap, that is 1.6ns, the base unit of TOF times. [22]: TOF Overflow No stop signal arrived within timeout of 330ns. [23]: Invalid TOF Event If the TOF Event is measured in 1 anode (anodes 0-11 only) or two neighboring anodes (anodes 0-11 only) it is valid. Otherwise it is invalid, unless it was an underflow in which case the Underflow [21] counter is increased instead of this counter (i.e. an Underflow event is considered valid). Therefore, if the event is not an Underflow event, it will be invalid if one of these three situations is met: - hit in more than two anodes, or - hit in two non-neighbor anodes, or - no anodes hit at all. The latter is different to overflow events [22] which are considered valid. The Background anode is not considered in any of these calculations. [24]: Event Strobe The number of TOF Events completed, by a stop signal or over/underflow, usually the same as Start in Process Time [20]. Note that the look directions of logicals 12-24 cover the combined look directions of logicals 0-11. Anodes 0-3 will have an azimuth 180 degrees greater than anodes 4-11. For logicals 12-24 that cover all 12 anodes, the azimuth of anodes 4-11 will be used, but elevation will be -90 to +180 degrees, centered at +45 degrees (between anodes 5 and 6).
6689	6400	DATA_SIGMA	f	Counts/s	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*

Byte	Length (bytes)	Name	Fmt*	Units	Description
13089	6400	BACKGROUND	f	Counts/s	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
19489	6400	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
25889	6400	DIM1_E	f	eV/q	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
32289	6400	DIM1_E_UPPER	f	eV/q	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
38689	6400	DIM1_E_LOWER	f	eV/q	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
45089	6400	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
51489	6400	DIM2_ELEVATION_UPPER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
57889	6400	DIM2_ELEVATION_LOWER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
64289	6400	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
70689	6400	DIM2_AZIMUTH_DESPUN_UPPER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*
77089	6400	DIM2_AZIMUTH_DESPUN_LOWER	f	Degrees	*Same description as from* Table 71 *for JAD_L30_HRS_ION_ANY_CNT.*

6.2.10.5 Level 3 conversion of data for V01, V02 and V03

Moved to section 6.2.12 (as it's independent of version number).

6.2.11 Level 3 data files for file version 04+

This section (6.2.11) and sub-sections are only for Level 3 file version 04.
If you are after Level 3 file versions 01, 02 and 03, go to section 6.2.10.

The Level 3 data files are binary and have files ending in the extension .DAT. Accompanying them in the same directory are the label files with the same filename but the extension .LBL.

For example, the PDS file pairs will have the following paths in the Volume:

ROOT/DATA/*yyyy*/*yyyyddd*/*subdir*/JAD_L30_*aaa*_*bbb*_*ccc*_*uuu*_*yyyyddd*_V*nn*.DAT
ROOT/DATA/*yyyy*/*yyyyddd*/*subdir*/JAD_L30_*aaa*_*bbb*_*ccc*_*uuu*_*yyyyddd*_V*nn*.LBL

The format file (same filename minus the date part, but including the version number, with the extension .FMT) accompanying (and already listed within) the LBL files are usually found in the LABEL directory at the root of the volume – however it was decided to exclude this LABEL directory (and therefore exclude FMT files) as they are redundant and may be copy/pasted out of the LBL files. [FMT files are made locally for JADE file production, but do not get to the PDS.]

See section 3.1 for the explanation of JAD_L30_*aaa*_*bbb*_*ccc*_*uuu*_*yyyyddd*_V*nn*, and *subdir* is the subdirectory name given in Table 17.

There are currently 7 different Level 3 product types, see Table 61 for their sizes, but they are similar and they all have the same objects (per version) as a header. To save space in this document, Table 75 gives the 44-object header for the binary files for Level 3 products version 04(+), which is then used throughout. This is the same for all (with one object name exception), except the PACKETID (which can change within a product type for Level 3 data) that gives a different description for each packet, shown in blue, and the last 4 objects that have the same names but different sizes. The rest of the data product is the same format (floats) but may have different sizes. The UTC entries are not side by side due to PDS rules requiring multi-byte words to start on even byte boundaries, so are spaced by 1-byte words. The exception to the same object names in the header is that ion TOF and ion species have a SOURCE_SPECIES_REMAPPED object (= 0 if no remapping), whereas the ion logicals and electron products (that never have remapping) have an object called SPARE_ZEROS (=0). Since both these objects are one-byte unsigned integers, one may simply use SOURCE_SPECIES_REMAPPED for all 7 data products, as the value of zero (no remapping) is still appropriate for ion logicals or electrons.

Efforts were made to keep the objects as similar as possible (both in name and dimensions), as shown in Figure 12. Some may consider this redundant but this is deliberately done so that the same code may be used on different datasets. For example a 64 by 48 object may only contain 64 unique values that change with the 1st dimension during low rate science files, however during high rate science files both the 1st and 2nd dimension values change – since these objects are the same dimension the same code may then be used to analyze both high and low rate science files.

In order to have fewer products than level 2 had, like ones were grouped together to give just 7 products per unit, with the unit of counts per second being the base file, that files with other units are to be created from. Data from high voltage engineering and calibration modes are excluded from level 3 data, as they are not designed for science use (possibly with highly variable MCPs voltages for MCP tests).

Level 3 data should be scientifically useful data, however there is still an object called ISSUES. This is for occasions where the data is scientifically valid, but may not be similar to its neighbors. For instance, the data may be accumulating records over 30 second accumulation times, but the last record was during a mode change so there's only 13 seconds. The data for those 13 seconds are valid, but for consistency the end user may wish to disregard and only use the full 30 second data that's available. This ISSUES object allows such occurrences to be flagged easily.

If a level 2 high rate or low rates science record is unsuitable for science work, a level 3 record may still be created, however the DATA object will be replaced with MISSING_CONSTANT fill values. This is to allow a user to know that high or low rate data was deliberately excluded, but does exist in level 2 data. However when calibration mode data is excluded (as not for science), no equivalent record of fill values will exist in the level 3 data.

The MISSING_CONSTANT for the objects DATA, DATA_SIGMA, BACKGROUND and BACKGROUND_SIGMA is -999999 (not -1) in Level 3 version 04+ data (but not versions 01, 02 and 03). See section 0 for more details.

Table 62 lists the Level 3 products and which Level 2 products were used to get them. There are no high voltage engineering data in level 3 (no JAD_L20_HVE*), nor ion direct events (no JAD_*aaa*_ION_DER nor JAD_*aaa*_ION_DES).

As ion species records go in the same level 3 products, it is possible to have consecutive records with the same time stamp. The difference will be in the PACKETID that tells you which particular ion species that record is for. Likewise JAD_L30_LRS_ELC_ANY_CNT may contain records from any of the 3 electron sensors, however a given time will only ever have a record from one sensor record.

Note that the LBL/FMT files describe DATA, DATA_SIGMA, BACKGROUND, BACKGROUND_SIGMA, DIM1_*, DIM2_* and transformation matrices DESPUN_SC_TO_J2000 and J2000_TO_RTP as 2D or 3D containers (containers in containers that hold a scalar). If you read the object in as a 1D vector then it should be reformed by the user to a 2D or 3D array. The 1D ordering is based on c, in that the last dimension changes fastest, i.e. if a 1D array is x=[1, 2, 3, 4, 5, 6] and that should be a 3x2 array y, then:

y[0][0] = 1; y[0][1] = 2; y[1][0] = 3; y[1][1] = 4; y[2][0] = 5; y[2][1] = 6;

Object	Data Type	Total Number of Bytes	JAD_L30_HRS_ELC_ALL	JAD_L30_HRS_ELC_TWO	JAD_L30_LRS_ELC_ANY	JAD_L30_CAL_ELC_ALL	JAD_L30_HRS_ION_ANY	JAD_L30_LRS_ION_ANY	JAD_L30_CAL_ION_ANY	JAD_L30_HLS_ION_TOF	JAD_L30_CAL_ION_TOF	JAD_L30_HLS_ION_LOG	JAD_L30_CAL_ION_LOG
DIM0_UTC	char[21]	21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PACKETID	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM0_UTC_UPPER	char[21]	21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PACKET_MODE	int8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM0_UTC_LOWER	char[21]	21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PACKET_SPECIES	int8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ACCUMULATION_TIME	uint16[1]	2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DATA_UNITS	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOURCE_BACKGROUND	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOURCE_DEAD_TIME	uint8[1]	1											
SPARE_ZEROS	uint8[1]	1	✓	✓	✓	✓						✓	✓
SOURCE_SPECIES_REMAPPED	uint8[1]	1					✓	✓	✓	✓	✓		
SOURCE_MAG	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOURCE_JADE_METAKERNEL	int16[1]	2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOURCE_JADE_CALIB	int16[1]	2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FSW_VERSION	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LUT_VERSION	float(1)	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LUT_VERSION_SUB_LETTER	char[2]	2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LUT_SWEEP_TABLE	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FILE_VERSION	uint8[1]	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_R	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_R_UPPER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_R_LOWER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LAT	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LAT_UPPER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LAT_LOWER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LOCAL_TIME	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LOCAL_TIME_UPPER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_LOCAL_TIME_LOWER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_SYSIII_ELONG	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_SYSIII_ELONG_UPPER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_SYSIII_ELONG_LOWER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_POS_JUPITER_J2000XYZ	float[3]	12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_VEL_JUPITER_J2000XYZ	float[3]	12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_VEL_ANGULAR_J2000XYZ	float[3]	12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_SPIN_PERIOD	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_SPIN_PHASE	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_SPIN_PHASE_UPPER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SC_SPIN_PHASE_LOWER	float[1]	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DESPUN_SC_TO_J2000	float[3,3]	36	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
J2000_TO_JSSXYZ	float[3,3]	36	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
J2000_TO_JSSRTP	float[3,3]	36	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MCP_VOLTAGE	float	4, 8 or 12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ISSUES	uint32	4 or 8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TIMESTAMP_WHOLE	uint32	4 or 8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TIMESTAMP_SUB	uint16	2 or 4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DATA	float[64,n]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DATA_SIGMA	float[64,n]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BACKGROUND	float[64,n]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BACKGROUND_SIGMA	float[64,n]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM1_E	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM1_E_UPPER	float[64,m]	Depends											
DIM1_E_LOWER	float[64,m]	Depends											
DIM2_ELEVATION	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM2_ELEVATION_UPPER	float[64,m]	Depends											
DIM2_ELEVATION_LOWER	float[64,m]	Depends											
DIM2_AZIMUTH_DESPUN	float[64,m]	Depends	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIM2_AZIMUTH_DESPUN_UPPER	float[64,m]	Depends											
DIM2_AZIMUTH_DESPUN_LOWER	float[64,m]	Depends											
DIM3_TOF	float[n=93]	372								✓	✓		
DIM3_TOF_UPPER	float[n=93]	372											
DIM3_TOF_LOWER	float[n=93]	372											
TOF_WITH_START_OVERLOAD	float[64]	256								✓	✓		
TOF_WITH_START_OVERLOAD_SIGMA	float[64]	256								✓	✓		
TOF_TOO_SHORT	float[64]	256								✓	✓		
TOF_TOO_SHORT_SIGMA	float[64]	256								✓	✓		
TOF_TOO_LONG	float[64]	256								✓	✓		
TOF_TOO_LONG_SIGMA	float[64]	256								✓	✓		
MAG_VECTOR	float[3]	12	✓	✓	✓	✓							
ESENSOR	uint16[1]	2			✓								
Number of Objects			52	52	53	52	51	51	51	58	58	51	51

Figure 12: Breaking out the JADE Level 3 Version 04+ products in to the different PDS Objects to allow similarities to be drawn.
Grey columns represent calibration files for JADE operations use that will not go to the PDS. Blue text values do not need to be in level 3 files, but aids cross comparison with level 2 data, and red text are extra values that may be useful. m = n for all but TOF products, where m = 1 because of the 3rd TOF dimension.

The following table (over 10 pages) describes the header that is identical for all the following data version 04 (and is based on Level 3 Version 04 FMT files). The names and word type (int/float/etc.) for all level 3 version 04 data is also summarized in Figure 12. Any text in *red italics* is a note that is not in the FMT file, while any text in **blue boldface** may change depending on the product (usually just the product ID or species number). This color system will apply for format tables throughout the rest of section 6.2.

Table 75: Format of Level 3 data record header for Version 04+ (also for Level 5 binary files)

Byte	Length (bytes)	Name	Fmt*	Units	Description
1	21	DIM0_UTC	UTC string	Time	UTC timestamp at center (not start) of record. Format is yyyy-dddTHH:MM:SS.sss where yyyy = year, ddd = day of year, HH = hour, MM = minute, SS.sss = decimal seconds to millisecond resolution. Note: Duration of record can be found in S.I. seconds by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse this with the ACCUMULATION_TIME object, which is the number of spacecraft clock ticks for accumulation. While 1 tick is approximately 1 second, it is not identical.
22	1	PACKETID	uint8	None	Packet ID (DPID), Data Product Identifier **High Rate Science – Electron** **Two Electron sensors per record: E060 and E180.** **(This is the same data as for JAD_L30_HRS_ELC_ALL but with E300 data removed for a smaller file.)** **PACKETID = 142 (0x8E)**
23	21	DIM0_UTC_UPPER	UTC string	Time	0th Dimension of DATA: Time - upper limit. See DIM0_UTC for description.

Byte	Length (bytes)	Name	Fmt*	Units	Description
44	1	PACKET_MODE	int8	None	Packet Mode, describes type of data telemetry. -2 = HSK / Housekeeping Engineering (Level 2 only) -1 = HVE / High Voltage Engineering (Level 2 only) 0 = CAL / MCP Calibration Science (Level 2 only) 1 = LRS / Low Rate Science 2 = HRS / High Rate Science 3 = DRS / DeRived Science from LRS and/or HRS 127 = Unknown 254 = Wrong - but HSK, see below. (Level 2 only) 255 = Wrong - but HVE, see below. (Level 2 only) (Note, this could also be calculated via PACKETID.) If you have 254 or 255 then your code is incorrect, check you read a signed byte, rather than unsigned.
45	21	DIM0_UTC_LOWER	UTC string	Time	0th Dimension of DATA: Time - lower limit. See DIM0_UTC for description.
66	1	PACKET_SPECIES	int8	None	Packet Species, describes type of plasma data. -1 = electrons 0 = ion species 0, SP0 1 = ion species 1, SP1 2 = ion species 2, SP2 3 = ion species 3, SP3 4 = ion species 4, SP4 5 = ion species 5, SP5 6 = ion species 6, SP6 7 = ion species 7, SP7 8 = Sum of SP3, SP4 and SP5 9 = All ions /* or any ion, e.g., TOF and LOG */ 10 = Single ion species derived from TOF data 127 = Unknown 255 = Wrong - but electrons, see below. If you have 255 then your code is incorrect, check you read a signed byte, rather than unsigned.

<table>
<tr><th>Byte</th><th>Length (bytes)</th><th>Name</th><th>Fmt*</th><th>Units</th><th>Description</th></tr>
<tr><td>67</td><td>2</td><td>ACCUMULATION_TIME</td><td>uint16</td><td>SCLK ticks</td><td>Accumulation Time.
Number of seconds over which the data in this product was collected (Science Program).
Note: Duration of record can be found in S.I. seconds by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse this with the ACCUMULATION_TIME object, which is the number of spacecraft clock ticks for accumulation.
While 1 tick is approximately 1 second, it is not identical.
ACCUMULATION_TIME is left in spacecraft clock ticks to both aid matching with the level 2 data and to help filtering for data taken in a particular mode.</td></tr>
<tr><td>69</td><td>1</td><td>DATA_UNITS</td><td>uint8</td><td>None</td><td>Data units correspond to:
0 = All counts in the accumulation period
1 = All counts divided by number of views
2 = Counts per second
/* S.I. science units: */
3 = Differential Energy Flux [1/(m^2 sr s)]
4 = Differential Number Flux [1/(m^2 sr s J)]
5 = Phase Space Density [m^-6 s^3]
/* Convenient (non-S.I.) science units: */
6 = Differential Energy Flux [1/(cm^2 sr s)]
7 = Differential Number Flux [1/(cm^2 sr s keV)]
8 = Phase Space Density [cm^-6 s^3]
/* As new products are developed this list will increase */
/* If a number is not listed, */
/* try a LBL/FMT file from a recent date. */
255 = Unknown.</td></tr>
</table>

Byte	Length (bytes)	Name	Fmt*	Units	Description
70	1	SOURCE_BACKGROUND	uint8	None	Source of Background values (see BACKGROUND object) that have been removed from the DATA object. 0 = None: No background has been removed 1 = Background anode (electron sensors only) 2 = Background anode (JADE-I only) 3 = Derived from Background anode : Method 1: Background coefficients are time independent. See file in CALIB directory for description. 4 = Derived from Background anode : Method 2: Background coefficients are per orbit. See file in CALIB directory for description. /* As new background removal methods are developed this list will increase */ 255 = Unknown.
71	1	SOURCE_SPECIES_REMAPPED **Or** SPARE_ZEROS	uint8	None	**Source of ion remapping for ION Species/TOF data products:** **0 = None: Data has not been remapped on the ground.** **255 = Unknown.** *A new object for Level 3 Version 4 files (TOF and ion species only), and for V04 files SOURCE_SPECIES_REMAPPED = 0 always. The JADE team has no current plans to remap the data, but this otherwise spare byte would allow us to track any remapping if carried out. Replaced SOURCE_DEAD_TIME from the version 01, 02 and 03 files.* **Or** **Spare Zeroes. Always zero.** **PDS3 format required a padding byte, e.g., a 4-byte integer/float will always start on the 1st or 5th or 9th or 13th... byte of the record**. *A new object for Version 4 files (Electrons or ion logicals only). Replaced SOURCE_DEAD_TIME from the version 01, 02 and 03 files.*

Byte	Length (bytes)	Name	Fmt*	Units	Description
72	1	SOURCE_MAG	uint8	None	Source of MAG data Except case 0 and 1, PAYLOAD (pl) co-ordinate MAG files were used at 1s (or 2s if no 1s) resolution. 0 = None: No MAG data in this product. 1 = From Juno JADE's Level 2 files. (From spacecraft and therefore uncalibrated.) This is independent to JADE Level 2 version number as it does not change with versions. [Note MAG data in JADE files may be affected by the Juno time stutter.] 3n = Juno's MAG's Level 3 version n calibrated files, e.g., 34 means version 4, so: 30 = From Juno MAG's Level 3 version 00 quicklook payload files. (These are temporary files not in PDS.) 31 = From Juno MAG's Level 3 version 01 calibrated payload files. 32 = From Juno MAG's Level 3 version 02 calibrated payload files. Likewise, 33 to 39 being Level 3 version 3 to 9. 255 = Unknown. If you see a number not listed above, there may be later versions of MAG data - find the latest available LBL file for this product and see what that has listed.
73	2	SOURCE_JADE_METAKERNEL	int16	None	The JADE SPICE metakernel used to get the time, position, velocity, orientation and transformation objects in this file. The metakernel lists the many individual spice kernels used, which are archived by NAIF and not in this PDS volume. The JADE SPICE metakernel may be found in the CALIB directory of this PDS volume, with filenames of: JAD_L30_SPICE_METAKERNEL_nnnnn.TXT where nnnnn is the SOURCE_JADE_METAKERNEL object number (with leading zeros and positive). If any of the kernels within the metakernel are not reconstructed (but reference or predicted) for the time in question, this value will be negative. Within the JADE PDS archive this value should always be positive.

Byte	Length (bytes)	Name	Fmt*	Units	Description
75	2	SOURCE_JADE_CALIB	int16	None	The JADE calibration files list used to convert the engineering units of Level 2 data to the scientific units in this file. Similar to the SPICE metakernel list, this lists the many individual calibration files used, each of which may be found in the CALIB directory on this PDS volume. This list may be found in the CALIB directory of this PDS volume, with filenames of: JAD_L30_CALIB_LIST_nnnnn.TXT where nnnnn is the SOURCE_JADE_CALIB object number (with leading zeros and positive). If any of the calibration files listed are not final at the time in question, this value will be negative. (Newer calibration files will have a higher version and simply be listed in a newer SOURCE_JADE_CALIB file.) Within the JADE PDS archive this value should always be positive. However, a version 00 file (for team use or uploaded to JSOC, not PDS) may have negative values with predicted positions/orientations/transformations.
77	4	FSW_VERSION	f	None	Flight Software version used. Number should be to 2 decimal places, with rounding. e.g., 4.00, 4.10, 4.20. i.e., 4.1999998 means 4.20.
81	4	LUT_VERSION	f	None	LUT (Look Up Table) Version used on JADE. Number should be to 2 decimal places, with rounding. e.g., 4.00, 4.10, 4.20. i.e., 4.1999998 means 4.20.

Byte	Length (bytes)	Name	Fmt*	Units	Description
85	2	LUT_VERSION_SUB_LETTER	string	None	The letter (if any) associated with the energy table used at the time of this record -- -> No sub letter for this LUT Version -A -> Sub letter is A for this LUT Version -B -> Sub letter is B for this LUT Version -C -> Sub letter is C for this LUT Version etc. For instance, the energy table files are in the CALIB directory of this PDS volume, with names like: LUT_4_00_ENERGY_V01.CSV (LUT_VERSION 4.00, no sub letter) or LUT_5_01_K_ENERGY_V01.CSV (LUT_VERSION 5.01, sub letter K).
87	1	LUT_SWEEP_TABLE	uint8	None	**The sweep tables the ion sensor used. A level 2 packet will report this as 0-3, However, it requires 2 packets (a ping and a pong) to make a level 3 record: either 0 and 1, or 2 and 3. Therefore, a value of 1 (= 01) means sweep tables 0 and 1 were used, while a value of 23 means sweep tables 2 and 3 were used. This object can only have the value of 1 or 23.** *There is a different description for low rate electrons, and different again for high rate electrons.*
88	1	FILE_VERSION	uint8	None	The version number of the file this record came from. e.g., if you loaded file JAD_L30_LRS_ION_ANY_CNT_2016240_V04.DAT then FILE_VERSION = 4. [FILE_VERSION = 0 is never in the PDS, but is used by the JADE team prior to having required calibrations.]
89	4	SC_POS_R	f	R_J	Juno radial distance at time DIM0_UTC, from Jupiter, in units of Jupiter Radii (Rj). (1 Rj = 71492.0 km) [Values may be greater than VALID_MAXIMUM during cruise to Jupiter before primary mission.]

Byte	Length (bytes)	Name	Fmt*	Units	Description
93	4	SC_POS_R_UPPER	f	R_J	Juno radial distance at time DIM0_UTC_UPPER, from Jupiter, in units of Jupiter Radii (Rj). (1 Rj = 71492.0 km) SC_POS_R_UPPER could be smaller or larger than SC_POS_R, depending if moving inbound or outbound. [Values may be greater than VALID_MAXIMUM during cruise to Jupiter before primary mission.]
97	4	SC_POS_R_LOWER	f	R_J	Juno radial distance at time DIM0_UTC_LOWER, from Jupiter, in units of Jupiter Radii (Rj). (1 Rj = 71492.0 km) SC_POS_R_LOWER could be smaller or larger than SC_POS_R, depending if moving inbound or outbound. [Values may be greater than VALID_MAXIMUM during cruise to Jupiter before primary mission.]
101	4	SC_POS_LAT	f	Degrees	Juno Latitude at time DIM0_UTC, in both the IAU_JUPITER and JUNO_JSS frames, in units of degrees. (0 = Equatorial) (JUNO_JSS is a despun version of IAU_JUPITER, hence they have identical latitudes.)
105	4	SC_POS_LAT_UPPER	f	Degrees	Juno Latitude at time DIM0_UTC_UPPER, in both the IAU_JUPITER and JUNO_JSS frames, in units of degrees. (0 = Equatorial) SC_POS_LAT_UPPER could be smaller or larger than SC_POS_LAT. (JUNO_JSS is a despun version of IAU_JUPITER, hence they have identical latitudes.)
109	4	SC_POS_LAT_LOWER	f	Degrees	Juno Latitude at time DIM0_UTC_LOWER, in both the IAU_JUPITER and JUNO_JSS frames, in units of degrees. (0 = Equatorial) SC_POS_LAT_LOWER could be smaller or larger than SC_POS_LAT. (JUNO_JSS is a despun version of IAU_JUPITER, hence they have identical latitudes.)

Byte	Length (bytes)	Name	Fmt*	Units	Description
113	4	SC_POS_LOCAL_TIME	f	Hours	Juno's (jovian) Local Time at time DIM0_UTC, in units of hours. 00 = Midnight 06 = Dawn 12 = Noon 18 = Dusk
117	4	SC_POS_LOCAL_TIME_UPPER	f	Hours	Juno's (jovian) Local Time at time DIM0_UTC_UPPER, in units of hours. 00 = Midnight 06 = Dawn 12 = Noon 18 = Dusk
121	4	SC_POS_LOCAL_TIME_LOWER	f	Hours	Juno's (jovian) Local Time at time DIM0_UTC_LOWER, in units of hours. 00 = Midnight 06 = Dawn 12 = Noon 18 = Dusk
125	4	SC_POS_SYSIII_ELONG	f	Degrees	Juno's (jovian) SYSIII (East) Longitude at time DIM0_UTC, in units of degrees.
129	4	SC_POS_SYSIII_ELONG_UPPER	f	Degrees	Juno's (jovian) SYSIII (East) Longitude at time DIM0_UTC_UPPER, in units of degrees.
133	4	SC_POS_SYSIII_ELONG_LOWER	f	Degrees	Juno's (jovian) SYSIII (East) Longitude at time DIM0_UTC_LOWER, in units of degrees.
137	12	SC_POS_JUPITER_J2000XYZ	f	km	Juno position from Jupiter in J2000 Cartesian co-ordinates [x,y,z] (units km). [Values may be outside of VALID_MIN/MAX range (~140Rj) during cruise to Jupiter before primary mission.]
149	12	SC_VEL_JUPITER_J2000XYZ	f	km/s	Juno Velocity with respect to Jupiter in J2000 Cartesian co-ordinates [Vx,Vy,Vz] (units km/s).
161	12	SC_VEL_ANGULAR_J2000XYZ	f	rads/s	Juno Angular Velocity in Cartesian co-ordinates [AVx,AVy,AVz] (units radians/s). (This is calculated with the SPICE ckgpav command where ref=J2000. SPICE defines it as 'This is the axis about which the reference frame tied to the instrument is rotating in the right-handed sense.')
173	4	SC_SPIN_PERIOD	f	Seconds	Juno spin period (seconds). This is not useful during spacecraft maneuvers.
177	4	SC_SPIN_PHASE	f	Seconds	Juno's spin phase at time DIM0_UTC, in units of degrees.

Byte	Length (bytes)	Name	Fmt*	Units	Description
181	4	SC_SPIN_PHASE_UPPER	f	Seconds	Juno's spin phase at time DIM0_UTC_UPPER, in units of degrees.
185	4	SC_SPIN_PHASE_LOWER	f	Seconds	Juno's spin phase at time DIM0_UTC_LOWER, in units of degrees.
189	36	DESPUN_SC_TO_J2000	f	None	Rotation matrix from despun spacecraft co-ordinates to J2000. This is a 3x3 matrix, but if read in as a 1x9 stream then the 1D stream is [a,b,c, d,e,f, g,h,i] and the 2D matrix would be [a,b,c d,e,f g,h,i]
225	36	J2000_TO_JSSXYZ	f	None	Rotation matrix from J2000 co-ordinates to JSS xyz (JSS = Jupiter-De-Spun-Sun, see SIS for details). This is a 3x3 matrix, but if read in as a 1x9 stream then the 1D stream is [a,b,c, d,e,f, g,h,i] and the 2D matrix would be [a,b,c d,e,f g,h,i]
261	36	J2000_TO_JSSRTP	f	None	Rotation matrix from J2000 co-ordinates to JSS RTP, where RTP is Jupiter centered right handed R-Theta-Phi. (JSS = Jupiter-De-Spun-Sun, see SIS for details.) This is a 3x3 matrix, but if read in as a 1x9 stream then the 1D stream is [a,b,c, d,e,f, g,h,i] and the 2D matrix would be [a,b,c d,e,f g,h,i]
297		MCP_VOLTAGE			The last 4 objects of this header all start at byte 297 and have the same names, but three different sizes depending on the JADE product. For Level 3 and Level 5 products: - For the ion products go to Table 76. - For the HRS electrons (all) go to Table 77, or Table 78 for HRS electrons (two). - For the LRS electrons go to Table 79.
		ISSUES			
		TIMESTAMP_WHOLE			
		TIMESTAMP_SUB			

Fmt* is shortened for the table and is decoded in PDS format as: f = PC_REAL (float), uint8/uint16/uint32 are = one/two/four-byte LSB_UNSIGNED_INTEGER and int8/int16/int32 = one/two/four byte LSB_INTEGER.

Table 76: Format of Level 3 data record subheader for Level 3 ion products for V04+ (also for Level 5 binary files)

Byte	Length (bytes)	Name	Fmt*	Units	Description
297	4	MCP_VOLTAGE	f	Volts	MCP Voltage on sensor.
301	8	ISSUES	uint32	None	Issues or potential issues in this data record. *[Two values for ions as this is the ISSUES object from both the ping and pong level 2 packets used to create this record.]...* The rest is a direct copy of the Level 2 ISSUES object, see Table 45 for description.
309	8	TIMESTAMP_WHOLE	uint32	Ticks	Timestamps (Whole Second) of JADE Level 2 packets used to make this Level 3 record. (Both the ping and pong level 2 packets.) Note: Timestamp is in Spacecraft clock ticks.
317	4	TIMESTAMP_SUB	uint16	Subticks	Timestamps (Subsecond) of JADE Level 2 packets used to make this Level 3 record. (Both the ping and pong level 2 packets.) **A value of 0 could be valid or a MISSING_CONSTANT, but should only be treated as a MISSING_CONSTANT if TIMESTAMP_WHOLE is also 0.**

Table 77: Format of Level 3 data record subheader for JAD_L30_HRS_ELC_ALL_* for V04+

Byte	Length (bytes)	Name	Fmt*	Units	Description
297	12	MCP_VOLTAGE	f	Volts	MCP Voltages on the three electron sensors, E060, E180 and E300 respectively.
309	4	ISSUES	uint32	None	Issues or potential issues in this data record. … The rest is a direct copy of the Level 2 ISSUES object, see Table 45 for description.
313	4	TIMESTAMP_WHOLE	uint32	Ticks	Timestamp (Whole Second) of JADE Level 2 packet used to make this Level 3 record. Note: Timestamp is in Spacecraft clock ticks.
317	2	TIMESTAMP_SUB	uint16	Subticks	Timestamp (Subsecond) of JADE Level 2 packet used to make this Level 3 record. **A value of 0 could be valid or a MISSING_CONSTANT, but should only be treated as a MISSING_CONSTANT if TIMESTAMP_WHOLE is also 0.**

Table 78: Format of Level 3 data record subheader for JAD_L30_HRS_ELC_TWO_* for V04+ (also for Level 5 binary files)

Byte	Length (bytes)	Name	Fmt*	Units	Description
297	8	MCP_VOLTAGE	f	Volts	MCP Voltages on the two electron sensors in this product, E060 and E180 respectively.
305	4	ISSUES	uint32	None	Issues or potential issues in this data record. … The rest is a direct copy of the Level 2 ISSUES object, see Table 45 for description.
309	4	TIMESTAMP_WHOLE	uint32	Ticks	Timestamp (Whole Second) of JADE Level 2 packet used to make this Level 3 record. Note: Timestamp is in Spacecraft clock ticks.
313	2	TIMESTAMP_SUB	uint16	Subticks	Timestamp (Subsecond) of JADE Level 2 packet used to make this Level 3 record. **A value of 0 could be valid or a MISSING_CONSTANT, but should only be treated as a MISSING_CONSTANT if TIMESTAMP_WHOLE is also 0.**

Table 79: Format of Level 3 data record subheader for JAD_L30_LRS_ELC_ANY_* for V04+ (also for Level 5 binary files)

Byte	Length (bytes)	Name	Fmt*	Units	Description
297	4	MCP_VOLTAGE	f	Volts	MCP Voltage on sensor.
301	4	ISSUES	uint32	None	Issues or potential issues in this data record. … The rest is a direct copy of the Level 2 ISSUES object, see Table 45 for description.
305	4	TIMESTAMP_WHOLE	uint32	Ticks	Timestamp (Whole Second) of JADE Level 2 packet used to make this Level 3 record. Note: Timestamp is in Spacecraft clock ticks.
309	2	TIMESTAMP_SUB	uint16	Subticks	Timestamp (Subsecond) of JADE Level 2 packet used to make this Level 3 record. **A value of 0 could be valid or a MISSING_CONSTANT, but should only be treated as a MISSING_CONSTANT if TIMESTAMP_WHOLE is also 0.**

In general, the rest of the format for the different products have the same object names (see Figure 12), however their size (byte length) and start bytes will differ. The descriptions are also much the same when they have the same object name, with only DATA really changing (text that may alter between products is shown in **blue boldface**).

6.2.11.1 Electron Data for V04+

6.2.11.1.1 JAD_L30_HRS_ELC_ALL_CNT_* for V04+

The electron product for high rate science is PACKETID 0x8E and includes data from all three electron sensors.
The DATA object is 2-D, 64 energies x 48 look directions, and is described in Table 80, and continues over the next 3 pages.

This product is a combination of look directions from all 3 JADE-E sensors, but E300 was turned off in 2016, hence those anodes that would have been from E300 are populated with the MISSING_CONSTANT (-1) value.
If E300 was off on a given day, from Level 3 Version 04 we no longer generate the daily file, since JAD_L30_HRS_ELC_TWO_CNT files (still generated, see Table 81) have the exact same information. Thus if there is a JAD_L30_HRS_ELC_ALL_CNT_*V04 file, all 3 JADE-E sensors were on.

Table 80: Format of Level 3 data records for JAD_L30_HRS_ELC_ALL_CNT for V04+

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 77 *for bytes 1 to 318.*					
319	12288	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 48 Look Directions. [Note: E300 was turned off in 2016, so the last 16 look directions (32-47) are usually populated with the MISSING_CONSTANT value of -999999.]
12607	12288	DATA_SIGMA	f	Counts/s	DATA_SIGMA 1-sigma uncertainties on values in object DATA, such that true value = DATA +/- DATA_SIGMA. See DATA entry above for size information.
24895	12288	BACKGROUND	f	Counts/s	Background value removed from DATA. If you wish to do your own background removal, add this object to DATA then you can remove a background via your own method. See the SOURCE_BACKGROUND object for the background method used per record. The background values here were generated from a background anode or JADE's own ground method, or are all zeros if no background was removed.

Byte	Length (bytes)	Name	Fmt*	Units	Description
37183	12288	BACKGROUND_SIGMA	f	Counts/s	BACKGROUND_SIGMA 1-sigma uncertainties on values in object BACKGROUND, such that true value = BACKGROUND +/- BACKGROUND_SIGMA. See **BACKGROUND entry above for size information.**
49471	12288	DIM1_E	f	eV/q	1st Dimension of DATA: Energy (center) in eV/q.
61759	12288	DIM2_ELEVATION	f	Degrees	2nd Dimension of DATA: Spacecraft elevation - center value. Spacecraft elevation (degs) is analogous to latitude on a sphere. In spacecraft xyz co-ords: +z is equivalent to elevation = +90 degs -z is equivalent to elevation = -90 degs (The communication dish is directed along +z) xy-plane at z = 0 is equivalent to elevation = 0. Note, 2nd dimension is really look direction which has an elevation and azimuth; hence two objects describe this: DIM2_ELEVATION and DIM2_AZIMUTH_DESPUN.

Byte	Length (bytes)	Name	Fmt*	Units	Description
74047	12288	DIM2_AZIMUTH_DESPUN	f	Degrees	2nd Dimension of DATA: Despun S/C azimuth - center value. Spacecraft azimuth (degs) is analogous to longitude on a sphere. In spacecraft xyz co-ords: +x is equivalent to azimuth = 0 degs +y is equivalent to azimuth = 90 degs -x is equivalent to azimuth = 180 degs -y is equivalent to azimuth = 270 degs +x is equivalent to azimuth = 360 degs +y is equivalent to azimuth = 450 degs The 'Despun' azimuth angle varies because Juno spins, where azimuth = 0 is defined as +x when spin phase equals zero (e.g. despun x-z plane contains the ECLIPJ2000 north). The relationship between despun azimuth and spin phase (which decreases during a spin) is simply: Despun Azimuth = 360 degrees - Spin Phase Note, 2nd dimension is really look direction which has an elevation and azimuth; hence two objects describe this: DIM2_ELEVATION and DIM2_AZIMUTH_DESPUN.
86335	12	MAG_VECTOR	f	nT	MAG vector in nT, 3 components [X, Y, Z] MAG range is +/- 16 G, hence limits. This xyz coordinate system is despun spacecraft; see the definitions of DIM2_ELEVATION and DIM2_AZIMUTH: +X is when [azimuth, elevation] = [0, 0] degrees, +Y is when [azimuth, elevation] = [90, 0] degrees, +Z is when elevation = 90 degrees.

6.2.11.1.2 JAD_L30_HRS_ELC_TWO_CNT_* for V04+

This is a repeat of the JAD_L30_HRS_ELC_ALL_CNT_* file, but with E300 data removed to provide a smaller (but still large) file, thus only contains E060 and E180 data. This product was introduced when it was decided not to use sensor E300 in flight operations, however the HRS electron data packet would still return zeros for E300.
The DATA object is 2-D, 64 energies x 32 look directions (rather than 48 look directions), and is described in Table 81.

Table 81: Format of Level 3 data records for JAD_L30_HRS_ELC_TWO_CNT for V04+

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 78 *for bytes 1 to 314.*					
315	8192	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 32 Look Directions.
8507	8192	DATA_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
16699	8192	BACKGROUND	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
24891	8192	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
33083	8192	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
41275	8192	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
49467	8192	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
57659	12	MAG_VECTOR	f	nT	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*

6.2.11.1.3 JAD_L30_LRS_ELC_ANY_CNT_* for V04+

The electron products for low rate science are PACKETIDs 0x68, 0x6A and 0x6B, and includes data from one electron sensor per record (only one sensor is on at any given time).
The DATA object is 2-D, 64 energies x 48 look directions, and is described in Table 82.
Practically there are only two differences between this and the JAD_L30_HRS_ELC_ALL_CNT_* file:

1) The MCP_VOLTAGE object is a singular value here (for the one sensor) as opposed to 3 values for the HRS case (one for each of the sensors). This in turn makes the start byte of all following objects 8 bytes earlier in the LRS product compared to the HRS product. The description of MCP_VOLTAGE in the FMT file is slightly different to reflect this.
2) This product has an extra object at the end; called ESENSOR that states which of the three sensors is in use (60, 180 or 300). This does not exist in the HRS product as the data array always includes all three sensors.

So the only difference between tables Table 68 and Table 70 are the first column byte values are offset by 8 (as indicated in the first red row), and Table 70 has the ESENSOR product at the end.

If using FSW4.00 (which was April 2015 only) data for this product (cruise solar wind only, no Jupiter science use) all *DIM2_AZIMUTH_DESPUN* values were replaced with the fill value 65535 due to the reverse anode mapping bug (see section 6.2.9.1.4).

Table 82: Format of Level 3 data records for JAD_L30_LRS_ELC_ANY_CNT for V04+

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 79 *for bytes 1 to 310.*					
311	12288	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 48 Look Directions.
12599	12288	DATA_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
24887	12288	BACKGROUND	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
37175	12288	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
49463	12288	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
61751	12288	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
74039	12288	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
86327	12	MAG_VECTOR	f	nT	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
86339	2	ESENSOR	uint16	None	ESENSOR - which one of the three electron sensors is this record for. Values can only be 60, 180 or 300 for electron sensor E060, E180 or E300 respectively. Note: each sensor also has a different PACKETID.

6.2.11.2 Ion Species Data for V04+

6.2.11.2.1 JAD_L30_HRS_ION_ANY_CNT_* for V04+

The ion species products for high rate science cover PACKETIDs 0x80-0x87. Each ion species has its own packet; therefore several packets of different species may have the same time stamp. The DATA object is 2-D, 64 energies x 12 look directions, and is described in Table 83.

Table 83: Format of Level 3 data records for JAD_L30_HRS_ION_ANY_CNT for V04+

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 76 *for bytes 1 to 320.*					
321	3072	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 12 Look Directions.
3393	3072	DATA_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
6465	3072	BACKGROUND	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
9537	3072	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
12609	3072	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
15681	3072	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
18753	3072	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*

6.2.11.2.2 JAD_L30_LRS_ION_ANY_CNT_* for V04+

The ion species products for low rate science (PACKETID 0x60-0x67). Each ion species has its own packet; therefore several packets of different species may have the same time stamp.
The DATA object is 2-D, 64 energies x 78 look directions, and is described in Table 84.
The basic format of this file is identical to the HRS counterpart, except there are 78 look directions here instead of 12. As such the start byte and lengths change, but the object names and descriptions are the same (except for the description of the DATA object).

Table 84: Format of Level 3 data records for JAD_L30_LRS_ION_ANY_CNT for V04+

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 76 *for bytes 1 to 320.*					
321	19968	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 78 Look Directions.
20289	19968	DATA_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
40257	19968	BACKGROUND	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
60225	19968	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
80193	19968	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
100161	19968	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
120129	19968	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*

6.2.11.3 Ion Time of Flight Data for V04+

6.2.11.3.1 JAD_L30_HLS_ION_TOF_CNT_* for V04+

The ion time of flight products for high and low rate science, covering PACKETIDs 0x69 and 0x89.
The DATA object is 3-D, 64 energies x 1 look direction x 93 TOF channels, and is described in Table 85 (over 2 pages). This product usually has 96 TOF channels with the last 3 having special meanings, but for level 3 data the last 3 channels have been removed and given their own objects within this file.

This product is usually considered to be a 2 dimensional array of energy by TOF channel. However all other JADE data is Energy by look direction, so to keep things similar, this product is a 3 dimensional array of 64 energies by 1 look direction by 93 TOF channels. There is only 1 look direction, but given the ion instrument covers 270 degrees field of view in elevation over the 12 anodes, and this product sums all 12 anodes, this leads to some interesting azimuth and elevation numbers. The DIM2_AZIMUTH objects will use the respective azimuth of anodes 4-11 (anodes 0-3 azimuths would normally be 180 degrees from those). However DIM2_ELEVATION will range from -90 to +180 degrees (spanning 270 degrees) with a center value of +45 degrees. As such, elevation of +90 to +180 is being used to describe the contribution of anodes 3, 2, 1 and 0 that are technically covering elevations of +90 down to 0 degrees but with an azimuth 180 degrees different.

The object names (and descriptions, DATA description excepted) are identical to the other level 3 ion products, but with 6 TOF only objects on the end. (Text that may alter between products is shown in **blue boldface**, e.g. version number of files should match the version number of the DAT files.)

Table 85: Format of Level 3 data records for JAD_L30_HLS_ION_TOF_CNT for V04+

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 76 *for bytes 1 to 320.*					
321	23808	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 1 Look Direction x 93 bins. These bins are expressed as a duration in seconds in object DIM3_TOF, and for more details see the TOF_CHANNEL_TO_SECONDS_HLC_V**04**.CSV file in the CALIB directory of this PDS archive. The Level 2 data had 96 bins, those last 3 are now objects TOF_WITH_START_OVERLOAD, TOF_TOO_SHORT and TOF_TOO_LONG respectively.
24129	23808	DATA_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
47937	23808	BACKGROUND	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
71745	23808	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
95553	256	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
95809	256	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
96065	256	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
96321	372	DIM3_TOF	f	Seconds	3rd Dimension of DATA: Time Of Flight - center value. (Seconds)
96693	256	TOF_WITH_START_OVERLOAD	f	Counts/s	TOF with start overload: Counts/Second A signal pulse that is too strong (above a threshold) in the electronics. Multiple start-overloads that occur within a 330ns event window are counted each time in the Logicals Start Overload, but only once here.
96949	256	TOF_WITH_START_OVERLOAD_SIGMA	f	Counts/s	TOF with start overload uncertainty: Counts/Second 1-sigma uncertainties on values in object TOF_WITH_START_OVERLOAD such that true value = TOF_WITH_START_OVERLOAD +/- TOF_WITH_START_OVERLOAD_SIGMA. See TOF_WITH_START_OVERLOAD entry above for size information.

Byte	Length (bytes)	Name	Fmt*	Units	Description
97205	256	TOF_TOO_SHORT	f	Counts/s	TOF too short: Counts/Second TOF underflow: Count of TOF measurements that did not timeout, but resulted in a measurement smaller than the sensor could measure.
97461	256	TOF_TOO_SHORT_SIGMA	f	Counts/s	TOF too short uncertainty: Counts/Second 1-sigma uncertainties on values in object TOF_TOO_SHORT such that true value = TOF_TOO_SHORT +/- TOF_TOO_SHORT_SIGMA. See TOF_TOO_SHORT entry above for size information.
97717	256	TOF_TOO_LONG	f	Counts/s	TOF too long: Counts/Second TOF overflow: Count of TOF measurements that resulted in no stop signal arriving within 330ns of the start signal.
97973	256	TOF_TOO_LONG_SIGMA	f	Counts/s	TOF too long uncertainty: Counts/Second 1-sigma uncertainties on values in object TOF_TOO_LONG such that true value = TOF_TOO_LONG +/- TOF_TOO_LONG_SIGMA. See TOF_TOO_LONG entry above for size information.

6.2.11.4 Ion Logicals Data for V04+

6.2.11.4.1 JAD_L30_HLS_ION_LOG_CNT_* for V04+

The ion logicals products for high and low rate science, covering PACKETID 0x6C and 0x8C. The DATA object is 2-D, 64 energies x 25 logicals (each with variable look directions), and is described in Table 86 (over 3 pages).

Given the ion instrument covers 270 degrees field of view in elevation, this leads to some interesting azimuth and elevation numbers, as elevation can range from -90 to +180 degrees; see the descriptions below. e.g. if Azimuth is 200 degrees and elevation is 100 degrees, that's equivalent to an azimuth of 20 (200-180) degrees and an elevation of 80 (180-100) degrees. That is anode 0 will have an azimuth 180 degrees from anode's 7, which is described in the DIM2 objects, however the logicals that combine all individual anodes the DIM2 values will use the azimuth from anodes 4-11 for all, but the elevations range will be -90 to +180 degrees.

The 25 logical counters here are the same as for level 2 data.

The PDS ion logicals datasets do not have a background removed from DATA, hence SOURCE_BACKGROUND should always be 0, and BACKGROUND and BACKGROUND_SIGMA should always be zero too.

Table 86: Format of Level 3 data records for JAD_L30_HLS_ION_LOG_CNT for V04+

Byte	**Length (bytes)**	**Name**	**Fmt***	**Units**	**Description**
See Level 2 binary header from Table 75 *and* Table 76 *for bytes 1 to 320.*					
321	6400	DATA	f	Counts/s	DATA: Counts/Second 64 Energy x 25 Logicals. The 25 Logical counters are: [0]: Anode 0 Events [1]: Anode 1 Events [2]: Anode 2 Events … [10]: Anode 10 Events [11]: Anode 11 Events [12]: Background Events The above 13 logicals are raw count hits, independent of whether a TOF Event has begun. Adjacent and Non-Adjacent hits will be counted in both anodes. As such, anode counts can exceed All Stops [15] counts. The Background anode [12] is not included in Adjacent and Non-Adjacent calculations. [13]: Start Overload Start signal exceeds threshold level. [14]: All Starts Independent of whether a TOF Event has begun, usually starts a TOF Event. [15]: All Stops Independent of whether a TOF Event has begun, usually ends a TOF Event. If an event is seen on multiple anodes this counter is still only incremented once, therefore this is usually less than the sum of anodes 0 to 11. The Background anode is not included in All Stops, just anodes 0 to 11. [16]: Non-Adjacent Anodes This is either two non-neighbor anodes (anodes 0-11 only), or more than 2 anodes. [17]: Adjacent Anodes A count hit was measured in neighboring anodes; other products (e.g. Ion Species) will assign this to just the lower anode. [18]: Stop without Start A stop signal was received before a TOF Event was initiated by a start. *Continues on next page.*

Byte	Length (bytes)	Name	Fmt*	Units	Description
					Continues from previous page. [19]: Dual Start A TOF Event had started but one or more other start signals were received before a stop signal or the TOF Event overflowed. [20]: Start in Process Time The number of TOF Events started, can be less than All Starts [14]. [21]: TOF Underflow Received a stop event before 1 tap, that is 1.45ns, the base unit of TOF times. [22]: TOF Overflow No stop signal arrived within timeout of 330ns. [23]: Invalid TOF Event If the TOF Event is measured in 1 anode (anodes 0-11 only) or two neighboring anodes (anodes 0-11 only) it is valid. Otherwise it is invalid, unless it was an underflow in which case the Underflow [21] counter is increased instead of this counter (i.e. an Underflow event is considered valid). Therefore, if the event is not an Underflow event, it will be invalid if one of these three situations is met: - hit in more than two anodes, or - hit in two non-neighbor anodes, or - no anodes hit at all. The latter is different to overflow events [22] which are considered valid. The Background anode is not considered in any of these calculations. [24]: Event Strobe The number of TOF Events completed, by a stop signal or over/underflow, usually the same as Start in Process Time [20]. For the above, a TOF Event is a start signal followed by either a stop signal or timeout. Note that the look directions of logicals 12-24 cover the combined look directions of logicals 0-11. Anodes 0-3 will have an azimuth 180 degrees greater than anodes 4-11. For logicals 12-24 that cover all 12 anodes, the azimuth of anodes 4-11 will be used, but elevation will be -90 to +180 degrees, centered at +45 degrees (between anodes 5 and 6).

Byte	Length (bytes)	Name	Fmt*	Units	Description
6721	6400	DATA_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
13121	6400	BACKGROUND	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
19521	6400	BACKGROUND_SIGMA	f	Counts/s	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
25921	6400	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
32321	6400	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
38721	6400	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*

6.2.11.5 Level 3 conversion of data for V04+

Moved to section 6.2.12 (as it is independent of version number).

6.2.12 Level 3 conversion of data

This section summarizes the equations used to convert from level 2 to level 3 data, in the order each is encountered in the production code used by the JADE team for level 3 files. Specific calibration values are listed in the CALIB directory of the Level 3 PDS volume in the `JAD_L30_CALIB_LIST_nnnnn.TXT` or `JAD_L30_CALIB_LIST_nnnnn.PDF` file, where the value for *nnnnn* is given the in level 3 SOURCE_JADE_CALIB object for each record. This file may point to other files in the CALIB directory.

Note that MISSING_CONSTANT (also known as fill) values may be present in the DATA object or any other level 2 or level 3 object, and all the codes have to check for these and react accordingly. For instance, if a DATA element is a MISSING_CONSTANT value, then the level 3 DATA element in counts per second (and DATA_SIGMA too) will also be set to the appropriate MISSING_CONSTANT value. This greatly complicates the coding, but is safer to propagate MISSING_CONSTANT values in DATA, energy or look directions when necessary.

6.2.12.1 Remove any records where the level 2 MCP_COMMANDED_VALUE object is zero

This may be just a few records as JADE is turning on, or during cruise it may be all values that day for some electron files (where essentially the electron sensor was 'off'). If no records are left, then do not make a level 3 file at all.

6.2.12.2 Correct timestamps affected by the Juno time stutter

Check the ISSUES object (bit 10) of each record to see if it was affected by the Juno time stutter. If so, remove 1 tick from the TIMESTAMP_WHOLE value (leave TIMESTAMP_SUB as is) and calculate the new UTC time. Set bit 10 of the ISSUES object to 0 (False) for that record, and set bit 5 to 1 (True) (to note that the time has been corrected).

6.2.12.3 Check for FSW 4.00 LRS/CAL ion species bug (early 2015 data only)

If the LRS/CAL ion species bug is present then all accumulation times are fill values, and the reported start time is actually the end time of the record (see ISSUES description). The level 3 data has been corrected for this; now reporting the correct start time (at least to within 1 ms) and accumulation time.

6.2.12.4 If ion data, merge ping and pong records to put all 64 energies in one record

In level 2 data a full sweep of energies for ion data must be split over two telemetry packets (and therefore two level 2 records) for transmission. Here we recombine them and re-order in increasing energy. There are four possible sweep tables for ion data, 0 to 3, with either 0 & 1 or 2 & 3 used, e.g. sweep tables will flip 0,1,0,1,0,1,0,1,… When JADE turns to a new telemetry mode (e.g. HRS to LRS) it may do so on any second even if a pair of sweep tables is not complete. Generally the first one it hits is called the ping, the second the pong, however that first one may be either an odd or even sweep table number. HRS data is the exception, where the ping is always a 0 or 2, and a pong always a 1 or 3 (this is so that HRS products can be compared on the same time boundaries). Not all pings may have a corresponding pong, nor all pongs a corresponding ping, either due to a data gap or a pair not being complete due to a mode change.

For HRS data, each ping or pong takes 1 spacecraft tick (ACCUMULATION_TIME = 1), so when merged ACCUMULATION_TIME of those records is set to 2. For LRS or CAL data ACCUMULATION_TIME is unchanged as those already assume you're using both ping and pong. This is important for the conversion to counts/second later.

6.2.12.5 Remap energy steps in to ascending eV/q order

The Level 2 data records list energy steps in the order they were taken (e.g. every 2nd step up, then back down with every 2nd step of the ones that were missed going up, such that they interleave over a whole record). This step order is re-ordered in to one of increased eV/q. (See the `LUT_`*`m`*`_`*`nn`*`_ENERGY_V`*`vv`*`.CSV` files for the eV/q values of the Level 2 files, which allows one to work out how to remap them to be increasing in eV/q.)

6.2.12.6 Convert MCP_COMMANDED_VALUE to units of volts

The Level 2 data has object MCP_COMMANDED_VALUE which is a digital value that needs to be converted to a Level 3 object MCP_VOLTAGE (in volts).
The equations to use are listed in the `JAD_L30_CALIB_LIST_`*`nnnnn`*`.TXT` file.
Note that if MCP_COMMANDED_VALUE = 0 then MCP_VOLTAGE = 0V, but for non-zero values use the equations in the above file.

6.2.12.7 Use SPICE to calculate auxiliary information

Use the latest (at time of processing) version metakernel file from the CALIB directory to find position, orientation, velocities, spin period and co-ordinate transformation matrices, as well as the start/center/stop spin-phase values for each record. (Metakernel files are named `JAD_L30_SPICE_METAKERNEL_`*`nnnnn`*`.TXT` where each level 3 files has an object named SOURCE_JADE_METAKERNEL that contains the *`nnnnn`* value of the metakernel used to create that particular record.)

6.2.12.8 Apply any dead time corrections (V01, V02 and V03 only).

Currently there is no known reason to correct for dead time.
Since no dead time correction is applied (record object SOURCE_DEAD_TIME = 0 in all Level 3 version 01, 02 or 03 files), the object SOURCE_DEAD_TIME was removed from Level 3 version 04(+) files.

6.2.12.9 Convert level 2 counts to a more representative value and work out uncertainties

Level 2 DATA are all integers, which required some rounding, whereas Level 3 DATA are floats, so here we swap out the integers for the floats they would have been (using the lossy LUT compression tables) and calculate an uncertainty for each value to populate DATA_SIGMA. This is a much more complex procedure than you would like; hence we do it for you. It is explained in great detail in the CALIB directory file DATA_UNCERTAINTY_EQNS_V*nn*.PDF (See `JAD_L30_CALIB_LIST_`*nnnnn*`.TXT` file for which V*nn* you should use for each record.)

6.2.12.10 Convert Data and uncertainties to counts per second.

At this point the DATA (and DATA_SIGMA) objects are in units of counts per accumulation or counts per view, both need converting to counts per second.

It should be noted that during each spacecraft clock tick (assumed to be 1 S.I. second, although technically not true, but extremely close) the electron sensors sweep 64 energy steps per tick, while the ion sensor sweeps 32 energy steps (such that the ion sensor requires 2 seconds to measure all 64 energy steps). For both electron and ion sensors, the first 2 ms at each step is a settling period where no data is recorded while the voltage stabilizes. Hence the (1/64 - 0.002) and (1/32 - 0.002) terms in the following equations.

So while these conversion are quoted as to counts/second, they are technically counts per spacecraft clock tick. During flight so far, 1 spacecraft clock tick is within 0.0002% of 1 S.I. second, so assuming 1 tick equal 1 second is suitable (see the SPICE SCLKSCET kernel for variations in ticks compared to S.I. seconds).

For counts per accumulation products (where the level 2 object is total counts measured over a time period) the conversion to counts per second is as follows:

For electron HRS and electron CAL data:

$$C/sec = \frac{C}{ACCUMULATION\_TIME\left(\frac{1}{64} - 0.002\right)}$$

(Eqn. 1)

Note that for HRS electron data, ACCUMULATION_TIME = 1, so this simplifies to:

$$C/sec = \frac{C}{\frac{1}{64} - 0.002}$$

(Eqn. 2)

For **merged ping-pong** ion data (a record with 64 energy steps) for all TOF and LOG data, and HRS ion species:

$$C/sec = \frac{2C}{ACCUMULATION\_TIME\left(\frac{1}{32} - 0.002\right)}$$

(Eqn. 3)

Note that for HRS **merged ping-pong** ion species data, ACCUMULATION_TIME = 2, so this simplifies to:

$$C/sec = \frac{C}{\frac{1}{32} - 0.002}$$

(Eqn. 4)

The uncertainty (DATA_SIGMA) of the DATA is calculated with similar equations to give the uncertainty counts per second.

For rate products the level 2 data returns a 'per view' average value so that the conversion of data is simpler, but the uncertainty is much more complex.
For electron LRS data:

$$C/sec = \frac{C/View}{\left(\frac{1}{64} - 0.002\right)} \qquad \textit{(Eqn. 5)}$$

For ion species data (merged ping-pong or not) for both LRS and CAL data:

$$C/sec = \frac{C/View}{\left(\frac{1}{32} - 0.002\right)} \qquad \textit{(Eqn. 6)}$$

Similar calculations are done for any level 2 background anodes used for a background later, although beware that some rate products have a background object that is total counts (and may be compressed differently to its corresponding DATA object), so must be converted accordingly. However the Level 3 TOF products TOF_WITH_START_OVERLOAD, TOF_TOO_SHORT and TOF_TOO_LONG and their uncertainties are calculated as above for TOF data.

6.2.12.11 Remove non-DATA elements from DATA arrays

For high rate and calibration electron data the DATA array is size 64x51, where 64x48 is the actual data, and 64x3 are the background anodes. The level 3 DATA object is just the 64x48 array, and the background data are discarded (unless used later as an input to the BACKGROUND object).
Likewise the level 2 ion TOF DATA object is size 64x1x96 where the actual data is 64x1x93, and the last 3 have special meaning. The Level 3 TOF DATA object is size 64x1x93, and the others are given their own objects in the TOF record: TOF_WITH_START_OVERLOAD, TOF_TOO_SHORT and TOF_TOO_LONG – each of size 64.

6.2.12.12 Remove an appropriate background to 'clean' the dataset.

Remove a background and/or clean the data (e.g. remove false co-incidences ("ghost peaks" of other ion species) from ion species products), and remove that from the level 3 data object. Calculate the uncertainty on that background, and propagate that uncertainty with the data uncertainty to replace DATA_SIGMA.

For Level 3 version 01 and 02 files we are not removing a background nor 'cleaning' the data, hence the BACKGROUND object is zeros (and likewise BACKGROUND_SIGMA is zeros).

For Level 3 version 03 files (electrons only, there are no version 3 ion files) a time-independent background is removed. (See next paragraph for how to find details of the version 03 background removal calculations.)

For Level 3 version 04 files a time-dependent (per orbit) background is removed from the electron, ion TOF and ion species data. No background is ever removed for ion logicals files.
(See the `JAD_L30_CALIB_LIST_00002.TXT` file more details on the background removal calculations used for Level 3 versions 03 and 04.)

6.2.12.13 Assign correct energy table to the data

Using the reported Look Up Table in Level 2 files (TABLES_VERSION object) and sweep table number (SWEEP_TABLE object), apply the corrected ground calibrated energy table to the data to fill DIM1_E objects. Version 00 files may use a temporary estimated energy table, but Version 01 onwards files will have the specific energy table used at their time. The energy tables are in the `CALIB/LUT_`*`m`*`_`*`nn`*`_ENERGY_V`*`vv`*`.CSV` or `CALIB/LUT_`*`m`*`_`*`nn`*`_`*`T`*`_ENERGY_V`*`vv`*`.CSV` files, with the particular one used listed in the `JAD_L30_CALIB_LIST_`*`nnnnn`*`.TXT` (each level 3 record lists *`nnnnn`* in the object SOURCE_JADE_CALIB). Alternatively, from level 3 V04 files, the objects LUT_VERSION and LUT_VERSION_SUB_LETTER were added to provide *`m`*`.`*`nn`* and *`T`* respectively (while LUT_SWEEP_TABLE is the sweep table number).

6.2.12.14 Populate azimuth and elevations angles in a despun frame.

For each look direction populate the DIM2_* azimuth and elevation angles, and upper and lower limits. This is despun so requires using SPICE to find the spin phase (from earlier) and adjust accordingly.

Check here for FSW3 or FSW 4.00 data (2015-Jan and before only), and if so, set the LRS electron data azimuths to fill values. This was because prior to FSW 4.10 (August 2015) the anode mapping to electron spin-phase sector was incorrectly reversed in flight software and cannot be reversed. See the ISSUES object for more information. This will not apply to any data at Jupiter (2016+).

Azimuth and elevation information is provided in the CALIB directory files: `ANODE_LOOK_ELC_DEFL_NONE_V`*`vv`* and `ANODE_LOOK_ION_DEFL_NONE_V`*`vv`*. (Version 01 DAT files use *`vv`* = `02`, *`vv`* = `01` is skipped.) When necessary for the product they are despun using spin phase. For HRS products, the values are given per energy step, accounting for the earlier 2ms settling time and smear introduced by the spacecraft spinning during each energy step. There may be a further correction to the elevation angle, which is dealt with later.

Ensure all azimuth angles (degrees) are positive, with the lower values being smaller than the center value, which itself is smaller than the upper value. It is possible some angles may be more than 360 degrees greater than the last, but in practice in *sin* or *cos* statements that has no effect.

6.2.12.15 If TOF data, Populate DIM3_* objects

Convert ground bin numbers 0-92 (HRS, LRS or CAL) to a real time range in seconds.
Onboard there are really 256 channels that are mapped down to the 96 (ground) bins of level 2 files, and these bins can have different widths.
The last three (onboard channels 253, 254 and 255 which map to ground bins 93, 94 and 95) have special meanings, and are separated out as their own objects in level 3 files. Note that onboard channels 248-252 inclusive are not mentioned; these are uses as padding onboard, so are always zero and never included in ground data.

See the CALIB file TOF_CHANNEL_TO_SECONDS_HLC_V*vv*.CSV for the final values in seconds, and see JAD_L30_CALIB_LIST_*nnnnn*.TXT for the conversion equations.
(For HVE TOF data (with 125 ground values (128 – 3 of special meaning) instead of 93 values) the situation is much the same, but use file TOF_CHANNEL_TO_SECONDS_HVE_V*vv*.CSV.)

6.2.12.16 If Electron data, despin MAG vector to same despun frame as the azimuths.

JADE Level 2 electron files have a MAG vector within them in spacecraft co-ordinates, ion data do not, so ion data do not have an included MAG_VECTOR object. For electron data, use the MAG_TIMESTAMP_WHOLE:MAG_TIMESTAMP_SUB spacecraft clock timestamp from Level 2 files to find the spin phase (sp) at that instant, and rotate MAG_VECTOR x and y components accordingly (z component does not change). If using SPICE to convert this MAG timestamp in to ephemeris time, be sure to use Juno's high precision clock code (NAIF_SPACECRAFT_ID = -61999) since MAG_TIMESTAMP_SUB is a two-byte value. Unfortunately, our Level 3 version 01 code for LRS electrons used the standard precision clock (one-byte value) which caused errors, and this was fixed in Level 3 version 02 files – see the CALIB directory for more information in file:

JADE_LEVEL3_V02_COMPARED_TO_V01_DESCRIPTION_V01.PDF

[Note that MAG_TIMESTAMP_WHOLE:MAG_TIMESTAMP_SUB if taken from JADE Level 2 files may be affected by the Juno time stutter, we do not attempt any correction for that.]

If the magnetic field is less than a commanded threshold (threshold was originally 200 nT, later changed to 25 nT, and could be altered in future) then the MAG_VECTOR is not provided (just zeros in Level 2, and in Level 3 is set to MISSING_CONSTANT), so MAG_VECTOR is only populated at low radial distances such as perijove passes.

For HRS Level 2 files there is no MAG_TIMESTAMP_SUB object, so it is assumed to be 00000 in Level 3 version 01 files, meaning the spin phase angle can be off by up to ~12 degrees (based on a 30s spin period). For Level 3 version 02 files, MAG_TIMESTAMP_SUB is assumed to be 32768 (= half a MAG_TIMESTAMP_WHOLE) so that the spin phase angle can be off by up to ± 6 degrees (rather than +12 and -0 degrees of version 01). [More explanation is provided in the CALIB file JADE_LEVEL3_V02_COMPARED_TO_V01_DESCRIPTION_V01.PDF]

The equations used to despin the MAG vector are simply:

```
L3.MAG_VECTOR_X = L2.MAG_VECTOR_X*COS(sp) - L2.MAG_VECTOR_Y*SIN(sp)
L3.MAG_VECTOR_Y = L2.MAG_VECTOR_X*SIN(sp) + L2.MAG_VECTOR_Y*COS(sp)
L3.MAG_VECTOR_Z = L2.MAG_VECTOR_Z
```

6.2.12.17 If HRS electron data at Jupiter, adjust the earlier elevation angles for the deflectors

The electron sensors have deflectors that are only active for HRS when the magnitude of the MAG_VECTOR is greater than a threshold magnitude, that was originally set to 200 nT (and later change to 25 nT, and could be altered again in future). As such this was first used during

PJ1 (2016-240) and never used during cruise (as the magnetic field was far below the (200 nT) threshold magnitude in the solar wind and magnetosphere). The deflectors adjust the elevation angle to track the magnetic field vector. This correction to DIM2_ELEVATION values (including upper and lower) needs to be done. The calibration equations used for this correction are given in the CALIB directory file `ANODE_LOOK_ELC_DEFL_EQNS_V`*`vv`*`.PDF`. (See file `JAD_L30_CALIB_LIST_`*`nnnnn`*`.TXT` for which `V`*`vv`*.) For all other times and modes (LRS/CAL or HRS with magnetic field magnitude under the threshold magnitude) the electron deflectors are off and the `ANODE_LOOK_ELC_DEFL_NONE_V`*`vv`* elevation angles are used.

[JADE-I does also have deflectors, however it was decided they would never be used.]

6.2.12.18 Level 2 records that do not get converted to Level 3

Now remove any records that are not worthy of becoming Level 3 files. Any Level 2 record with "ACCUM_TRUNCATION = 1 AND ACCUMULATION_TIME less than the rounded spin period" (LOG files excepted) , or MCP_COMMANDED_VALUE = 0, or MCP_NOT_AT_COMMANDED = 1 or TABLES_VERSION = -99.99 (=MISSING_CONSTANT) is excluded from becoming a Level 3 record and is removed. Records that have any MISSING_CONSTANT values in the DATA object are still converted.

HRS/CAL/HVE electron products are unique in that they use three sensors per record, rather than just one sensor per record for JADE-I or LRS electrons. For HRS electron Level 3 version 01 files, if any one of the three sensors had MCP_NOT_AT_COMMANDED = 1 then the entire record is excluded from level 3. However, it was realized that during certain situations (e.g. changing from LRS to HRS, or MCP dipping) it was possibly for an electron sensor to have MCP_NOT_AT_COMMANDED at 0 and the other sensors to be at 1. For HRS Level 3 version 02 files, if at least one of the three electron sensors had MCP_NOT_AT_COMMANDED = 0 then that record is kept for level 3 to keep the good data from that sensor(s), but the data for the other sensor(s) (with MCP_NOT_AT_COMMANDED = 1) are set to fill values. For such cases, this is marked in the ISSUES object of the record as "Bit 21". If all three electron sensors have MCP_NOT_AT_COMMANDED = 0 then the whole record is still excluded from Level 3 version 02 files. For more information see the ISSUES object description and the CALIB file: JADE_LEVEL3_V02_COMPARED_TO_V01_DESCRIPTION_V01.PDF
[In the PDS this only applies to HRS electron data; however for the JADE operations team, the CAL electron files are filtered similarly.]

For Level 3 records of JAD_L30_LRS_ELC_ANY and JAD_L30_LRS_ION_ANY (and JAD_L30_CAL_ION_ANY, not in PDS) if the ACCUMULATION_TIME of the record is less than the SC_SPIN_PERIOD (rounded to whole number) then remove it (as less than a whole spin for a spin product).

For Level 3 records of JAD_L30_HLS_ION_LOG (and JAD_L30_CAL_ION_LOG, not in PDS) if the ACCUMULATION_TIME of the record is an odd number then remove it (as missing a ping or a pong).

What records remain are written to a level 3 PDS compliant DAT file, if no records remain then no file is written.

6.2.12.19 Level 3 DATA and BACKGROUND MISSING_CONSTANT (fill) values

For Level 3 version 01, 02 and 03 data, the MISSING_CONSTANT value (also known as the fill value) for objects DATA, DATA_SIGMA, BACKGROUND and BACKGROUND_SIGMA are -1. In the initial design, we were never going to remove a background, so all counts would be positive, hence -1 was a reasonable MISSING_CONSTANT.

However, once we began removing background in version 03 files, it was possible for DATA to go negative. While our production code ensures that a background removed data value was never exactly -1, there was room for confusion as there were many valid DATA elements with counts near zero, positive or negative.

For Level 3 version 04(+) data, the MISSING_CONSTANT value for objects DATA, DATA_SIGMA, BACKGROUND and BACKGROUND_SIGMA are -999999. This value is so negative that no valid DATA elements would be near.

If this change of MISSING_CONSTANT value is an issue for your codes, you can easily do a find/where command in your code to turn any -999999s in these four objects back to -1.

6.2.12.20 Use SPICE to add position and orientation information

SPICE was used with reconstructed kernels to calculate the position, velocity, orientation and transformation matrix objects. (These objects all begin with SC_*, except for the transformation matrices DESPUN_SC_TO_J2000, J2000_TO_JSSXYZ and J2000_TO_JSSRTP). Predicted kernels may have been used for version 00 test files, but never for non-zero version numbers.

While there is a SC_POS_SYSIII_ELONG object in version 04 files, there is no System III LAT object since this value is identical to SC_POS_LAT that was already present in earlier versions.

6.2.12.21 The Jupiter De-Spun-Sun (JUNO_JSS) co-ordinate System

The Juno Jupiter De-Spun-Sun system is the primary Jovian co-ordinate system the JADE team uses (Figure 13) and is known as JUNO_JSS in the Juno SPICE frame kernel (file: fk/juno_v09.tf or latest version of this file). It is Jupiter-centered, with the Z-axis aligned with the Jovian spin axis but does not spin with the planet. The X-axis is in the plane containing the spin axis and the Jupiter-Sun vector, where the Sun position has be aberration corrected.

If:

J_{Omega} = unit vector of Jupiter spin axis

R_{JS} = unit vector of Jupiter to Sun line

Then:

$Z = J_{Omega}$

$Y = Z \times R_{JS}$

$X = Y \times Z$

If using SPICE, do not aberration correct (use `abcorr = 'none'` in SPICE commands such as `spkezr` or `spkpos`).

The R, Latitude and Local Time (LT) system is based on JUNO_JSS where R is the magnitude of the [x,y,z] vector (in planetary radii, R_J), Latitude is the inverse sine of z/R (degrees) and Local Time (0-24 hours, where 12 hours is along +X and 18 hours LT along +Y) is a different way of expressing *longitude* (degrees from +X, positive in the direction towards +Y), where:

$$\text{Local time} = [(\textit{longitude} + 180°) * 24/360\ ] \text{ MOD } 24 = [(\text{atan}(y,x) + \pi\ ) * 12/\pi\ ] \text{ MOD } 24$$

("atan" is the four quadrant inverse tangent of y and x expressed in radians.)

To calculate LT using SPICE, the command `et2lst` with `type = 'PLANETOCENTRIC'` will provide Local (solar) Time values, which is aberration corrected (`abcorr = 'LT+S'`) unlike the above JUNO_JSS *longitude* way. At Jupiter the difference between the methods is < 0.6 s LT, and since `et2lst` returns whole seconds only, both methods are equivalent in practice.

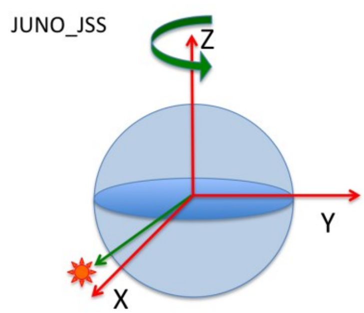


Figure 13: The Jupiter De-Spun-Sun (JUNO_JSS) co-ordinate system.

Note: System III latitude is identical to JUNO_JSS latitude, since they share the same Z axis.

6.2.13 Level 4 data files

There are no CODMAC Level 4 JADE products, as this is a level more designed for cameras than particle data (see Table 5). As such JADE goes from Level 3 to level 5 directly.

6.2.14 Level 5 data files

There are multiple dataset volumes for Level 5 data, some are binary and some are ASCII.

The Level 5 binary data files have files ending in the extension .DAT. Accompanying them in the same directory are the label files with the same filename but the extension .LBL.

For example, the PDS file pairs will have the following paths in the Volume:

ROOT/DATA/*yyyy*/*yyyyddd*/*subdir*/JAD_L50_*aaa_bbb_ccc_uuu_yyyyddd*_V*nn*.DAT
ROOT/DATA/*yyyy*/*yyyyddd*/*subdir*/JAD_L50_*aaa_bbb_ccc_uuu_yyyyddd*_V*nn*.LBL

The Level 5 ASCII data files have files ending in the extension .CSV. Accompanying them in the same directory are the label files with the same filename but the extension .LBL.

For example, the PDS file pairs will have the following paths in the Volume:

ROOT/DATA/*yyyy*/*yyyyddd*/JAD_L50_*aaa_bbb_ccc_ddd_ee_f_yyyyddd*_V*nn*.CSV
ROOT/DATA/*yyyy*/*yyyyddd*/JAD_L50_*aaa_bbb_ccc_ddd_ee_f_yyyyddd*_V*nn*.LBL

The format files (same filename minus the date part, but including the version number, with the extension .FMT) accompanying (and already listed within) the LBL files are usually found in the LABEL directory at the root of the volume – however it was decided to exclude this LABEL directory (and therefore exclude FMT files) as they are redundant and may be copy/pasted out of the LBL files. [FMT files are made locally for JADE file production, but do not get to the PDS.]

See section 3.1 for the explanation of JAD_L50_*aaa_bbb_ccc_uuu_yyyyddd*_V*nn* and JAD_L50_*aaa_bbb_ccc_ddd_ee_f_yyyyddd*_V*nn,* while *subdir* is the subdirectory name given in Table 18.

There are currently 8 different Level 5 product types, see Table 87 (binary) and Table 88 (ASCII) for their sizes.

Table 87: Size of a record of each Level 5 binary product, by version number

Version (*nn*)	Product	Bytes per record	Objects per record
01	JAD_L30_HLS_ION_TOF_DEF_V*nn*	96740	58
01	JAD_L30_HRS_ELC_TWO_DEF_V*nn*	65898	58
01	JAD_L30_HRS_ION_ANY_DEF_V*nn*	21872	57
01	JAD_L30_LRS_ELC_ANY_DEF_V*nn*	98664	59
01	JAD_L30_LRS_ION_ANY_DEF_V*nn*	140144	57

Table 88: Size of a record of each Level 5 ASCII product, by version number

Version (*nn*)	Product	Characters per record	Objects per record
01	JAD_L50_HLS_ELC_MOM_ISO_2D_ELECTRONS_V*nn*	233	23
01	JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES_V*nn*	377	27
01	JAD_L50_HLS_ION_MOM_ISO_3D_PROTONS_V*nn*	377	27

To save space in this document, Table 75 gives the 34-object header for the binary files for Level 5 products (i.e. same as for Level 3 Version 04+ header), which is then used throughout. This is the same for all, except the PACKETID (which can change within a product type for Level 5 data) that gives a different description for each packet, shown in blue, and the last 4 objects that have the same names but different sizes (and are provided in linked tables). The rest of the data product is the same format but may have different sizes. The UTC entries are not side by side due to PDS rules requiring multi-byte words to start on even byte boundaries, so are spaced by 1-byte words. [Apparently not a PDS rule after all, but files were already made.]

Efforts were made to keep the binary file objects as similar as possible (both in name and dimensions), as shown in Figure 14. Some may consider this redundant but this is deliberately done so that the same code may be used on different datasets. For example a 64 by 48 object may only contain 64 unique values that change with the 1st dimension during low rate science files, however during high rate science files both the 1st and 2nd dimension values change – since these objects are the same dimension the same code may then be used to analyze both high and low rate science files.

Efforts were also made to keep these formats as similar as possible to the Level 3 Version 04 format (e.g. you can compare Figure 12 with Figure 14). In fact, it is the same format with only a few new objects added on to the end (i.e. the exact same header of Table 75), and for the case of the electron files, the Level 3 MAG_VECTOR object has been removed (to be replaced several objects later with MAG_VECTOR_DESPUN). This was done so that code written to read in Level 3 Version 04 files can easily be adjusted to add the new level 5 objects on to the end. While the format is the same, there are some differences, i.e. For DATA, DATA_SIGMA, BACKGROUND and BACKGROUND_SIGMA objects, the VALID_MINIMIUM and VALID_MAXIMUM values, and UNIT are different, simply because the level 5 files are in different data units (i.e. differential energy flux instead of counts per second).

The Level 5 TOF files do not have a pitch angle object, since the TOF field of view is too large to make a pitch angle meaningful. Level 5 JADE-I ion species files also do not have a pitch angle object, as this needs to be calculated in the rest frame of the plasma, which is beyond a simple scientific unit translation of the data. However, the MAG vectors to go with each JADE record are included for the user's later use.

Data from high voltage engineering and calibration modes are excluded from level 5 data, as they are not designed for science use (possibly with highly variable MCPs voltages for MCP tests). Likewise Ion Logicals data is excluded from Level 5 data, as its aim is more operational or for calculating the ion background (already done in Level 3 data). The similarly operational Ion Direct Events files are also excluded from Level 5. Since JADE-E300 was not on while at Jupiter, there are no JAD_L50_HRS_ELC_ALL_DEF_V*nn* files.

The Level 5 files require calibrated MAG data to find a suitable magnetic field vector to align with the JADE record, and then to calculate pitch angles. The Level 3 1-second resolution payload MAG files from the PDS are used, which are released by PDS on 2-orbit cadences. In order to align, we look for the 1-second record closest to the center time of the JADE record. For low rate science JADE data, it must also be during the JADE record (i.e. JADE.DIM0_UTC_Lower <= MAG.UTC < JADE.DIM0_UTC_UPPER), otherwise a MISSING_CONSTANT (fill value) is used, and any pitch angles for that record are also MISSING_CONSTANT values. For high rate science JADE data, the MAG.UTC time stamp must be within 15s of the center time of the JADE record (i.e. within the same ~spin, JADE.DIM0_UTC – 15s <= MAG.UTC < JADE.DIM0_UTC + 15s), or else MISSING_CONSTANT values are used. See SOURCE_MAG object in each file for more specifics on the particular MAG file that was consulted.

It is also assumed that the MAG_VECTOR_DESPUN value (that takes the aligned MAG record and despins it in to our JADE frame) is representative of the whole ACCUMULATION_TIME period of the JADE record. If the magnetic field vector is fluctuating wildly during the JADE accumulation time, that is not captured in this level 5 JADE dataset, and the JADE generated pitch angles will no longer be reliable. It is up to the user to compare with PDS MAG data themselves to see if the magnetic field is stable enough over the period of time of JADE records accumulation times.

Table 89 lists the Level 5 products and which Level 3 products were used to get them.

Table 89: Mapping Level 3 data files to Level 5 binary data files

Level 3 Data Product	Path	Level 5 Data Product
JAD_L30_HRS_ELC_TWO_CNT	----------->	JAD_L50_HRS_ELC_TWO_DEF
JAD_L30_LRS_ELC_ANY_CNT	----------->	JAD_L50_LRS_ELC_ANY_DEF
JAD_L30_HRS_ION_ANY_CNT	----------->	JAD_L50_HRS_ION_ANY_DEF
JAD_L30_LRS_ION_ANY_CNT	----------->	JAD_L50_LRS_ION_ANY_DEF
JAD_L30_HLS_ION_TOF_CNT	----------->	JAD_L50_HLS_ION_TOF_DEF

As ion species records go in the same level 5 products, it is possible to have consecutive records with the same time stamp. The difference will be in the PACKETID that tells you which particular ion species that record is for. Likewise JAD_L50_LRS_ELC_ANY_DEF may contain records from any of the 3 electron sensors, however a given time will only ever have a record from one sensor record.

Note that the LBL/FMT files describe DATA, DATA_SIGMA, BACKGROUND, BACKGROUND_SIGMA, DIM1_*, DIM2_* and transformation matrices DESPUN_SC_TO_J2000 and J2000_TO_RTP as 2D or 3D containers (containers in containers that hold a scalar). If you read the object in as a 1D vector then it should be reformed by the user to a 2D or 3D array. The 1D ordering is based on c, in that the last dimension changes fastest, i.e. if a 1D array is x=[1, 2, 3, 4, 5, 6] and that should be a 3x2 array y, then:

y[0][0] = 1; y[0][1] = 2; y[1][0] = 3; y[1][1] = 4; y[2][0] = 5; y[2][1] = 6;

Object	Data Type	Total Number of Bytes	JAD_L50_HRS_ELC_TWO_DEF	JAD_L50_LRS_ELC_ANY_DEF	JAD_L50_HRS_ION_ANY_DEF	JAD_L50_LRS_ION_ANY_DEF	JAD_L50_HLS_ION_TOF_DEF
DIM0_UTC	char[21]	21	✓	✓	✓	✓	✓
PACKETID	uint8[1]	1	✓	✓	✓	✓	✓
DIM0_UTC_UPPER	char[21]	21	✓	✓	✓	✓	✓
PACKET_MODE	int8[1]	1	✓	✓	✓	✓	✓
DIM0_UTC_LOWER	char[21]	21	✓	✓	✓	✓	✓
PACKET_SPECIES	int8[1]	1	✓	✓	✓	✓	✓
ACCUMULATION_TIME	uint16[1]	2	✓	✓	✓	✓	✓
DATA_UNITS	uint8[1]	1	✓	✓	✓	✓	✓
SOURCE_BACKGROUND	uint8[1]	1	✓	✓	✓	✓	✓
SPARE_ZEROS	uint8[1]	1	✓	✓			
SOURCE_SPECIES_REMAPPED	uint8[1]	1			✓	✓	✓
SOURCE_MAG	uint8[1]	1	✓	✓	✓	✓	✓
SOURCE_JADE_METAKERNEL	int16[1]	2	✓	✓	✓	✓	✓
SOURCE_JADE_CALIB	int16[1]	2	✓	✓	✓	✓	✓
FSW_VERSION	float[1]	4	✓	✓	✓	✓	✓
LUT_VERSION	float(1)	4	✓	✓	✓	✓	✓
LUT_VERSION_SUB_LETTER	char[2]	2	✓	✓	✓	✓	✓
LUT_SWEEP_TABLE	uint8[1]	1	✓	✓	✓	✓	✓
FILE_VERSION	uint8[1]	1	✓	✓	✓	✓	✓
SC_POS_R	float[1]	4	✓	✓	✓	✓	✓
SC_POS_R_UPPER	float[1]	4	✓	✓	✓	✓	✓
SC_POS_R_LOWER	float[1]	4	✓	✓	✓	✓	✓
SC_POS_LAT	float[1]	4	✓	✓	✓	✓	✓
SC_POS_LAT_UPPER	float[1]	4	✓	✓	✓	✓	✓
SC_POS_LAT_LOWER	float[1]	4	✓	✓	✓	✓	✓
SC_POS_LOCAL_TIME	float[1]	4	✓	✓	✓	✓	✓
SC_POS_LOCAL_TIME_UPPER	float[1]	4	✓	✓	✓	✓	✓
SC_POS_LOCAL_TIME_LOWER	float[1]	4	✓	✓	✓	✓	✓
SC_POS_SYSIII_ELONG	float[1]	4	✓	✓	✓	✓	✓
SC_POS_SYSIII_ELONG_UPPER	float[1]	4	✓	✓	✓	✓	✓
SC_POS_SYSIII_ELONG_LOWER	float[1]	4	✓	✓	✓	✓	✓
SC_POS_JUPITER_J2000XYZ	float[3]	12	✓	✓	✓	✓	✓
SC_VEL_JUPITER_J2000XYZ	float[3]	12	✓	✓	✓	✓	✓
SC_VEL_ANGULAR_J2000XYZ	float[3]	12	✓	✓	✓	✓	✓
SC_SPIN_PERIOD	float[1]	4	✓	✓	✓	✓	✓
SC_SPIN_PHASE	float[1]	4	✓	✓	✓	✓	✓
SC_SPIN_PHASE_UPPER	float[1]	4	✓	✓	✓	✓	✓
SC_SPIN_PHASE_LOWER	float[1]	4	✓	✓	✓	✓	✓
DESPUN_SC_TO_J2000	float[3,3]	36	✓	✓	✓	✓	✓
J2000_TO_JSSXYZ	float[3,3]	36	✓	✓	✓	✓	✓
J2000_TO_JSSRTP	float[3,3]	36	✓	✓	✓	✓	✓
MCP_VOLTAGE	float	4, 8 or 12	✓	✓	✓	✓	✓
ISSUES	uint32	4 or 8	✓	✓	✓	✓	✓
TIMESTAMP_WHOLE	uint32	4 or 8	✓	✓	✓	✓	✓
TIMESTAMP_SUB	uint16	2 or 4	✓	✓	✓	✓	✓
DATA	float[64,n]	Depends	✓	✓	✓	✓	✓
DATA_SIGMA	float[64,n]	Depends	✓	✓	✓	✓	✓
BACKGROUND	float[64,n]	Depends	✓	✓	✓	✓	✓
BACKGROUND_SIGMA	float[64,n]	Depends	✓	✓	✓	✓	✓
DIM1_E	float[64,m]	Depends	✓	✓	✓	✓	✓
DIM2_ELEVATION	float[64,m]	Depends	✓	✓	✓	✓	✓
DIM2_AZIMUTH_DESPUN	float[64,m]	Depends	✓	✓	✓	✓	✓
DIM3_TOF	float[n=93]	372					✓
~~TOF_WITH_START_OVERLOAD~~	~~float[64]~~	~~256~~					~~✓~~
~~TOF_WITH_START_OVERLOAD_SIGMA~~	~~float[64]~~	~~256~~					~~✓~~
~~TOF_TOO_SHORT~~	~~float[64]~~	~~256~~					~~✓~~
~~TOF_TOO_SHORT_SIGMA~~	~~float[64]~~	~~256~~					~~✓~~
~~TOF_TOO_LONG~~	~~float[64]~~	~~256~~					~~✓~~
~~TOF_TOO_LONG_SIGMA~~	~~float[64]~~	~~256~~					~~✓~~
~~MAG_VECTOR~~	~~float[3]~~	~~12~~	~~✓~~	~~✓~~			
ESENSOR	uint16[1]	2		✓			
MAG_UTC	char[21]	21	✓	✓	✓	✓	✓
SOURCE_JADE_LEVEL3_VERSION_INPUT	uint8[1]	1	✓	✓	✓	✓	✓
SOURCE_JADE_GFXEFF_VERSION	uint8[1]	1	✓	✓	✓	✓	✓
MAG_RANGE	uint8[1]	1	✓	✓	✓	✓	✓
MAG_VECTOR_DESPUN	float[3]	12	✓	✓	✓	✓	✓
MAG_VECTOR_JSSRTP	float[3]	12	✓	✓	✓	✓	✓
DIM3_PITCH_ANGLES	float[64,n]	Depends	✓	✓			
Number of Objects			59	60	57	57	64

Figure 14: Breaking out the JADE Level 5 binary products in to the different PDS Objects to allow similarities to be drawn. (Crossed-out names are present in Level 3, but not Level 5.)
All objects before DIM3_TOF are the same size and in the same order as the Level 3 Version 04 data objects (see Figure 12*), with the exception that MAG_VECTOR was removed. (m = n for all but TOF products, where m = 1 because of the 3rd TOF dimension.)*

The binary header for Level 5 binary files is exactly the same as that for Level 3 Version 04+ files, as shown over 10 pages in Table 75, so will not be repeated here. The names and word type (int/float/etc.) for all level 5 data is also summarized in Figure 14. Any text in *red italics* is a note that is not in the FMT file, while any text in **blue boldface** may change depending on the product (usually just the product ID or species number). This color system will apply for format tables throughout the rest of section 6.2.

The binary footers are much the same too, except starting at different byte numbers (and the TOF product excludes the pitch angle object.

Unit conversion from Counts/s to Differential Energy Flux is just done by dividing the counts/s values by the 'Geometric Factor * Efficiency' value, which are given by equations. The version set of equations used is provided as object SOURCE_JADE_GFXEFF_VERSION that is provided per record in the binary files.

6.2.14.1 Electron Data

6.2.14.1.1 JAD_L50_HRS_ELC_TWO_DEF_*

The electron product for high rate science is PACKETID 0x8E. The DATA object is 2-D, 64 energies x 32 look directions (since JADE-E300 is excluded), and is described in Table 90 (over 3 pages).

Table 90: Format of Level 5 data records for JAD_L50_HRS_ELC_TWO_DEF

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 78 *for bytes 1 to 314.*					
315	8192	DATA	f	1/(m^2 sr s)	DATA: Differential Energy Flux (SI units) 64 Energy x 32 Look Directions.
8507	8192	DATA_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
16699	8192	BACKGROUND	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
24891	8192	BACKGROUND_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
33083	8192	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
41275	8192	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
49467	8192	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*

Byte	Length (bytes)	Name	Fmt*	Units	Description
57659	21	MAG_UTC	UTC string	Time	UTC timestamp of MAG record to be used for MAG related objects in this JADE record. Format is yyyy-dddTHH:MM:SS.sss where yyyy = year, ddd = day of year, HH = hour, MM = minute, SS.sss = decimal seconds to millisecond resolution. Generally this is the 1s resolution MAG data record closest to DIM0_UTC. For Low Rate Science data, this must be true: DIM0_UTC_LOWER <= MAG_UTC <= DIM0_UTC_UPPER or else MISSING_CONSTANT values are used in all MAG objects. For High Rate Science data, it must be within 30s (~1 spin) so that this must be true: DIM0_UTC - 15s <= MAG_UTC <= DIM0_UTC + 15s or else MISSING_CONSTANT values are used in all MAG objects.
57680	1	SOURCE_JADE_LEVEL3_VERSION_INPUT	uint8	None	Version number (nn) of Level 3 file JAD_L30_*_CNT_*Vnn file used as input to create this Level 5 file. e.g. if file JAD_L50_HRS_ELC_TWO_DEF_yyyyddd_V01 was generated from JAD_L30_HRS_ELC_TWO_CNT_yyyyddd_V04 then SOURCE_JADE_LEVEL3_VERSION_INPUT = 4 (and FILE_VERSION = 1).
57681	1	SOURCE_JADE_GFXEFF_VERSION	uint8	None	Version number of the Geometric Factor * EFFiciency calculation for this record (time dependent).
57682	1	MAG_RANGE	uint8	None	MAG instrument range (0-6). From the MAG PDS files, described as: [MAG] Instrument dynamic range identifier at time of the sample. See the Level 3 MAG SIS for further detail.

Byte	Length (bytes)	Name	Fmt*	Units	Description
57683	12	MAG_VECTOR_DESPUN	f	nT	Despun MAG vector in nT, 3 components [X, Y, Z] at time MAG_UTC. MAG range is +/- 16 G (= 1600000 nT), hence limits. This xyz coordinate system is despun spacecraft; see the definitions of DIM2_ELEVATION and DIM2_AZIMUTH: +X is when [azimuth, elevation] = [0, 0] degrees, +Y is when [azimuth, elevation] = [90, 0] degrees, +Z is when elevation = 90 degrees.
57695	12	MAG_VECTOR_JSSRTP	f	nT	MAG vector in JSS spherical components [Br, Bth, Bphi] in nT at time MAG_UTC, and is intended as a guide for context only. If you wish to do science with the MAG data, please use the PDS MAG datasets, and not this down-sampled vector used for the JADE data. This vector is identical to System III spherical components, since the JUNO_JSS frame has the same spin-axis as System III (IAU_JUPITER).
57707	8192	DIM3_PITCH_ANGLES_DIM1	f	Degrees	Pitch Angles of each element of the DATA object. The MAG vector provided in MAG_VECTOR_DESPUN was used, and it is assumed that that MAG vector is constant over the whole accumulation period.

6.2.14.1.2 JAD_L50_LRS_ELC_ANY_DEF_*

The electron products for low rate science are PACKETIDs 0x68, 0x6A and 0x6B, and includes data from one electron sensor per record (only one sensor is on at any given time).
The DATA object is 2-D, 64 energies x 48 look directions, and is described in Table 82.
Practically there are only two differences between this and the JAD_L50_HRS_ELC_TWO_DEF_* file:

1) The MCP_VOLTAGE object is a singular value here (for the one sensor) as opposed to multiple values for the HRS case (one for each of the sensors). This in turn makes the start byte of all following objects 4 bytes earlier in the LRS product compared to the HRS TWO product. The description of MCP_VOLTAGE in the FMT file is slightly different to reflect this.
2) This product has an extra object, called ESENSOR, that states which of the three sensors is in use (60, 180 or 300). This does not exist in the HRS product as the data array always includes multiple sensors.

If using FSW4.00 (which was April 2015 only) data for this product (cruise solar wind only, no Jupiter science use) all *DIM2_AZIMUTH_DESPUN* values were replaced with the fill value 65535 due to the reverse anode mapping bug (see section 6.2.9.1.4).

Table 91: Format of Level 5 data records for JAD_L50_LRS_ELC_ANY_DEF

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 79 *for bytes 1 to 310.*					
311	12288	DATA	f	1/(m^2 sr s)	DATA: Differential Energy Flux (SI units) 64 Energy x 48 Look Directions.
12599	12288	DATA_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
24887	12288	BACKGROUND	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
37175	12288	BACKGROUND_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
49463	12288	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
61751	12288	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
74039	12288	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
86327	2	ESENSOR	uint16	None	ESENSOR - which one of the three electron sensors is this record for. Values can only be 60, 180 or 300 for electron sensor E060, E180 or E300 respectively. Note: each sensor also has a different PACKETID.
86329	21	MAG_UTC	UTC string	Time	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
86350	1	SOURCE_JADE_LEVEL3_VERSION_INPUT	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
86351	1	SOURCE_JADE_GFXEFF_VERSION	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
86352	1	MAG_RANGE	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
86353	12	MAG_VECTOR_DESPUN	f	nT	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
86365	12	MAG_VECTOR_JSSRTP	f	nT	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
86377	12288	DIM3_PITCH_ANGLES_DIM1	f	Degrees	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*

6.2.14.2 Ion Species Data

6.2.14.2.1 JAD_L50_HRS_ION_ANY_DEF_*

The ion species products for high rate science cover PACKETIDs 0x80-0x87. Each ion species has its own packet; therefore several packets of different species may have the same time stamp. The DATA object is 2-D, 64 energies x 12 look directions, and is described in Table 92.

Table 92: Format of Level 5 data records for JAD_L50_HRS_ION_ANY_DEF

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 76 *for bytes 1 to 320.*					
321	3072	DATA	f	1/(m^2 sr s)	DATA: Differential Energy Flux (SI units) 64 Energy x 12 Look Directions.
3393	3072	DATA_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
6465	3072	BACKGROUND	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
9537	3072	BACKGROUND_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
12609	3072	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
15681	3072	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
18753	3072	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
21825	21	MAG_UTC	UTC string	Time	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
21846	1	SOURCE_JADE_LEVEL3_VERSION_INPUT	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
21847	1	SOURCE_JADE_GFXEFF_VERSION	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
21848	1	MAG_RANGE	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
21849	12	MAG_VECTOR_DESPUN	f	nT	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
21861	12	MAG_VECTOR_JSSRTP	f	nT	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*

6.2.14.2.2 JAD_L50_LRS_ION_ANY_DEF_*

The ion species products for low rate science (PACKETID 0x60-0x67). Each ion species has its own packet; therefore several packets of different species may have the same time stamp.
The DATA object is 2-D, 64 energies x 78 look directions, and is described in Table 93.
The basic format of this file is identical to the HRS counterpart, except there are 78 look directions here instead of 12. As such the start byte and lengths change, but the object names and descriptions are the same (except for the description of the DATA object).

Table 93: Format of Level 5 data records for JAD_L50_LRS_ION_ANY_DEF

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 76 *for bytes 1 to 320.*					
321	19968	DATA	f	1/(m^2 sr s)	DATA: Differential Energy Flux (SI units) 64 Energy x 78 Look Directions.
20289	19968	DATA_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
40257	19968	BACKGROUND	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
60225	19968	BACKGROUND_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
80193	19968	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
100161	19968	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
120129	19968	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
140097	21	MAG_UTC	UTC string	Time	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
140118	1	SOURCE_JADE_LEVEL3_VERSION_INPUT	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
140119	1	SOURCE_JADE_GFXEFF_VERSION	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
140120	1	MAG_RANGE	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
140121	12	MAG_VECTOR_DESPUN	f	nT	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
140133	12	MAG_VECTOR_JSSRTP	f	nT	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*

6.2.14.3 Ion Time of Flight Data

6.2.14.3.1 JAD_L50_HLS_ION_TOF_DEF_*

The ion time of flight products for high and low rate science, covering PACKETIDs 0x69 and 0x89.
The DATA object is 3-D, 64 energies x 1 look direction x 93 TOF channels, and is described in Table 94. This product usually has 96 TOF channels with the last 3 having special meanings, but for level 3 data the last 3 channels have been removed and given their own objects within this file.

This product is usually considered to be a 2 dimensional array of energy by TOF channel. However all other JADE data is Energy by look direction, so to keep things similar, this product is a 3 dimensional array of 64 energies by 1 look direction by 93 TOF channels. There is only 1 look direction, but given the ion instrument covers 270 degrees field of view in elevation over the 12 anodes, and this product sums all 12 anodes, this leads to some interesting azimuth and elevation numbers. The DIM2_AZIMUTH objects will use the respective azimuth of anodes 4-11 (anodes 0-3 azimuths would normally be 180 degrees from those). However DIM2_ELEVATION will range from -90 to +180 degrees (spanning 270 degrees) with a center value of +45 degrees. As such, elevation of +90 to +180 is being used to describe the contribution of anodes 3, 2, 1 and 0 that are technically covering elevations of +90 down to 0 degrees but with an azimuth 180 degrees different.

DATA, DATA_SIGMA, BACKGROUND and BACKGROUND_SIGMA are in units of Differential Energy Flux (DEF). The other extra TOF_* objects seen in Level 3 TOF files (TOF_WITH_START_OVERLOAD, TOF_TOO_SHORT, TOF_TOO_LONG and their respective *_SIGMA) are not included in this Level 5 format, since it made no sense to convert these to units of DEF.

The object names (and descriptions, DATA description excepted) are identical to the other level 5 ion products, but with 6 TOF only objects on the end. (Text that may alter between products is shown in **blue boldface**, e.g. version number of files should match the version number of the DAT files.)

Table 94: Format of Level 5 data records for JAD_L50_HLS_ION_TOF_DEF

Byte	Length (bytes)	Name	Fmt*	Units	Description
See Level 2 binary header from Table 75 *and* Table 76 *for bytes 1 to 320.*					
321	23808	DATA	f	1/(m^2 sr s)	DATA: Differential Energy Flux (SI units) 64 Energy x 1 Look Direction x 93 bins. These bins are expressed as a duration in seconds in object DIM3_TOF, and for more details see the TOF_CHANNEL_TO_SECONDS_HLC_V04.CSV file in the CALIB directory of this PDS archive. The Level 2 data had 96 bins, those last 3 are now objects TOF_WITH_START_OVERLOAD, TOF_TOO_SHORT and TOF_TOO_LONG respectively.
24129	23808	DATA_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
47937	23808	BACKGROUND	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
71745	23808	BACKGROUND_SIGMA	f	1/(m^2 sr s)	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
95553	256	DIM1_E	f	eV/q	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
95809	256	DIM2_ELEVATION	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
96065	256	DIM2_AZIMUTH_DESPUN	f	Degrees	*Same description as from* Table 80 *for JAD_L30_HRS_ELC_ALL_CNT.*
96321	372	DIM3_TOF	f	Seconds	3rd Dimension of DATA: Time Of Flight - center value. (Seconds)
96693	21	MAG_UTC	UTC string	Time	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
96714	1	SOURCE_JADE_LEVEL3_VERSION_INPUT	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
96715	1	SOURCE_JADE_GFXEFF_VERSION	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
96716	1	MAG_RANGE	uint8	None	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
96717	12	MAG_VECTOR_DESPUN	f	nT	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*
96729	12	MAG_VECTOR_JSSRTP	f	nT	*Same description as from* Table 90 *for JAD_L50_HRS_ELC_TWO_DEF.*

6.2.14.4 Moments Data (ASCII)

We use comma separated files, in particular the PD3 SPREADSHEET object format, which allows one object to have several elements (i.e. one velocity object goes over 3 columns). This means the number of columns in a file may be more than the number of objects. For these files we use a comma followed by a space (",\s") to separate columns, and end each line with the two bytes \r\n.

6.2.14.4.1 JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES_*

These files are 3D ion moments for heavy ions but provide an isotropic pressure and temperature (both the average of the diagonal of the pressure or temperature tensors). The data format for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES files is given in Table 95.

Table 95: Format of Level 5 data records for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES and JAD_L50_HLS_ION_MOM_ISO_3D_PROTONS

Field Number	Length (bytes)	Name	Fmt*	Units	Description
1	21	UTC	DATE	UTC	UTC timestamp at center (not start) of record. Format is yyyy-dddTHH:MM:SS.sss where yyyy = year, ddd = day of year, HH = hour, MM = minute, SS.sss = decimal seconds to millisecond resolution. Note: Duration of record can be found in seconds from ACCUMULATION_TIME. This record really covers the period starting at UTC - ACCUMULATION_TIME/2 (inclusive) and ending at UTC + ACCUMULATION_TIME/2 Technically, ACCUMULATION_TIME is in spacecraft clock ticks, where 1 tick is approximately 1 second, but is so close that, practically, we consider it as seconds.
2	3	SOURCE_JADE_LEVEL5_DEF_VERSION_INPUT	I3	None	The file version of the Level 5 DEF file used as input to calculate these moments. e.g. if file JAD_L50_LRS_ELC_ANY_DEF_2017112_V03.DAT was used then SOURCE_JADE_LEVEL5_DEF_VERSION_INPUT = 03.

Field Number	Length (bytes)	Name	Fmt*	Units	Description
3	3	INPUT_DATA_SELECTION	I3	None	Input Data Selection: which subset of input data was used to generate the moments? This is a simple look up table: 3 = ion species 3 only 4 = ion species 4 only 5 = ion species 5 only 34 = ion species 3 and 4 combined 45 = ion species 4 and 5 combined 60 = JADE-E060 only (electrons) 103 = ion species 3 only with TOF correction 104 = ion species 4 only with TOF correction 105 = ion species 5 only with TOF correction 134 = ion species 3 and 4 combined with TOF correction 145 = ion species 4 and 5 combined with TOF correction 180 = JADE-E180 only (electrons) 240 = JADE-E060 and JADE-E180 combined (electrons) 345 = ion species 3, 4 and 5 combined 999 = MISSING_CONSTANT = Unknown [Other entries may be added later as new techniques are explored/used. If your number is not listed here, try looking in the LBL file description of the latest file.] The TOF correction would account for false coincidence counts falling in other ion species datasets.
4	3	PACKET_MODE	I3	None	Packet Mode, describes type of data telemetry. 1 = LRS / Low Rate Science 2 = HRS / High Rate Science 127 = Unknown
5	5	ACCUMULATION_TIME	I5	s	Accumulation Time. Number of spacecraft clock ticks (assume seconds) over which the data in this product was collected. While 1 tick is approximately 1 second, it is not identical, but close enough that it is assumed to be.

Field Number	Length (bytes)	Name	Fmt*	Units	Description
6	3	SOURCE_BACKGROUND	I3	None	Source of Background values that have been removed from the DATA object. 0 = None: No background has been removed 1 = Background anode (electron sensors only) 2 = Background anode (JADE-I only) 3 = Derived from Background anode : Method 1: Background coefficients are time independent. See file in CALIB directory for description. 4 = Derived from Background anode : Method 2: Background coefficients are per orbit. See file in CALIB directory for description. /* As new background removal methods are developed this list will increase */ 255 = Unknown.
7	10	ISSUES	I10	None	Issues or potential issues in this data record. [Level 3 ion records have a ping and pong half, each with an ISSUES value. These two values have been merged with a bitwise OR to give a single value in this file.] *The ISSUES description is far too long to fit in this table, see* Table 45 *for the rest of this description.*
8	22	EV_PER_Q_RANGE	F10.3	eV/q	Energy Range of sensor(s) (eV/q) [lower, upper]. Each JADE sensor has its own energy range, and these do vary over time, with occasional significant changes of energy tables. This object is to give context to the moments, in particular, what energy range were moments calculated over. If two sensors data were combined, then this would reflect the merged energy table limits rather than of one particular sensor.
9	9	SC_POS_R	F9.3	R_J	Juno radial distance at time UTC, from Jupiter, in units of Jupiter Radii (Rj). (1 Rj = 71492.0 km) [Values may be greater than VALID_MAXIMUM during cruise to Jupiter before primary mission.]

Field Number	Length (bytes)	Name	Fmt*	Units	Description
10	9	SC_POS_LAT	F9.3	Degrees	Juno Latitude at time UTC, in both the IAU_JUPITER and JUNO_JSS frames, in units of degrees. (0 = Equatorial) (JUNO_JSS is a despun version of IAU_JUPITER, hence they have identical latitudes.)
11	9	SC_POS_LOCAL_TIME	F9.3	Hours	Juno's (jovian) Local Time at time UTC, in units of hours. 00 = Midnight 06 = Dawn 12 = Noon 18 = Dusk
12	9	SC_POS_SYSIII_ELONG	F9.3	Degrees	Juno's (jovian) SYSIII (East) Longitude at time UTC, in units of degrees.
13	1	DIMENSIONS	I1	None	Dimensionality of moments: are these calculated in 1D (=1), 2D (=2) or 3D (=3).
14	9	M	E9.3	amu	Mass of particle used for moments calculations in units of amu (atomic mass units). Valid minimum is 5.486E-04 amu, which is the mass of an electron (and why the E9.3 format was chosen), but the electron moment code actually used more precision with M_e of 0.00054857990907 amu.
15	5	Q	F5.2	e	Charge of particle used for moments calculations in units of e (elementary charge). i.e. an electron has a charge of -1, and a proton +1. e.g. For ions that are a mix of O+ and S++, we may use M = 24 and Q = 1.5, so that M/Q = 16.
16	3	NUM_LOOK_DIRS	I3	None	Number of Look Directions used in moments calculations. i.e. Low rate science ion species has 78 look directions, while 1D electron moments would only have 1 look direction.
17	10	N_CC	E10.3	cm^{-3}	Number Density in units of 1/cm^3.
18	10	N_SIGMA_CC	E10.3	cm^{-3}	Number Density Uncertainty in units of 1/cm^3.
19	34	V_JSSXYZ_KMPS	F10.3	km/s	Velocity Vector in the Cartesian JUNO_JSS (Jupiter-deSpun-Sun) frame in units of km/s. Three components are provided: [V_x, V_y, V_z]

Field Number	Length (bytes)	Name	Fmt*	Units	Description
20	34	V_JSSXYZ_SIGMA_KMPS	F10.3	km/s	Velocity Vector uncertainty in the Cartesian JUNO_JSS (Jupiter-deSpun-Sun) frame in units of km/s. Three components are provided: [V_sigma_x, V_sigma_y, V_sigma_z]
21	34	V_JSSRTP_KMPS	F10.3	km/s	Velocity Vector in the spherical JUNO_JSS (Jupiter-deSpun-Sun) frame in units of km/s. Three components are provided: [V_r, V_theta, V_phi]
22	34	V_JSSRTP_SIGMA_KMPS	F10.3	km/s	Velocity Vector uncertainty in the spherical JUNO_JSS (Jupiter-deSpun-Sun) frame in units of km/s. Three components are provided: [V_sigma_r, V_sigma_theta, V_sigma_phi]
23	10	PRESSURE_PA	E10.3	Pa	Isotropic pressure in units of Pascals.
24	10	PRESSURE_SIGMA_PA	E10.3	Pa	Isotropic pressure uncertainty in units of Pascals.
25	10	TEMP_EV	E10.3	eV	Isotropic temperature in units of eV.
26	10	TEMP_SIGMA_EV	E10.3	eV	Isotropic temperature uncertainty in units of eV.
27	3	QUALITY_FLAG	I3	None	Moments Quality Flag. To be determined for future versions: 255 = Unknown = MISSING_CONSTANT Currently this object is all values of 255.

6.2.14.4.2 JAD_L50_HLS_ION_MOM_ISO_3D_PROTONS_*

These files are 3D ion moments for protons but provide an isotropic pressure and temperature (both the average of the diagonal of the pressure or temperature tensors).

The data format for JAD_L50_HLS_ION_MOM_ISO_3D_PROTONS files is identical to the format of JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES files, so see Table 95.

6.2.14.4.3 JAD_L50_HLS_ELC_MOM_ISO_2D_ELECTRONS_*

These files are 2D moments for electrons, with an isotropic pressure and temperature. The number of pitch angle bins used for the 2D calculation (given in the NUM_LOOK_DIRS field) varies and is one of 1, 2, 3, 4, 6, 12 or 24, for pitch angle bin widths of 180, 90, 60, 45, 30, 15 or 7.5 degrees respectively. The files use the most pitch angle bins it can for each record, but neighboring records may have used different pitch angle bin widths.

The data format for JAD_L50_HLS_ELC_MOM_ISO_2D_ELECTRONS files is very similar to the format of JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES files, just without the 4 velocity objects, and are shown in Table 96 (over 2 pages).

Table 96: Format of Level 5 data records for JAD_L50_HLS_ELC_MOM_ISO_2D_ELECTRONS

Field Number	Length (bytes)	Name	Fmt*	Units	Description
1	21	UTC	DATE	UTC	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
2	3	SOURCE_JADE_LEVEL5_DEF_VERSION_INPUT	I3	None	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
3	3	INPUT_DATA_SELECTION	I3	None	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
4	3	PACKET_MODE	I3	None	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
5	5	ACCUMULATION_TIME	I5	s	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
6	3	SOURCE_BACKGROUND	I3	None	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
7	10	ISSUES	I10	None	Issues or potential issues in this data record. *The ISSUES description is far too long to fit in this table, see* Table 45 *for the rest of this description.*
8	22	EV_PER_Q_RANGE	F10.3	eV/q	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
9	9	SC_POS_R	F9.3	R_J	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*

Field Number	Length (bytes)	Name	Fmt*	Units	Description
10	9	SC_POS_LAT	F9.3	Degrees	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
11	9	SC_POS_LOCAL_TIME	F9.3	Hours	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
12	9	SC_POS_SYSIII_ELONG	F9.3	Degrees	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
13	1	DIMENSIONS	I1	None	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
14	9	M	E9.3	amu	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
15	5	Q	F5.2	e	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
16	3	NUM_LOOK_DIRS	I3	None	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
17	10	N_CC	E10.3	cm^{-3}	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
18	10	N_SIGMA_CC	E10.3	cm^{-3}	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
19	10	PRESSURE_PA	E10.3	Pa	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
20	10	PRESSURE_SIGMA_PA	E10.3	Pa	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
21	10	TEMP_EV	E10.3	eV	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
22	10	TEMP_SIGMA_EV	E10.3	eV	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*
23	3	QUALITY_FLAG	I3	None	*Same description as from* Table 95 *for JAD_L50_HLS_ION_MOM_ISO_3D_HEAVIES.*

Appendix A Support staff and cognizant persons

Table 97: Archive collection support staff

JADE team			
Name	**Address**	**Phone**	**Email**
Dr Rob Wilson JADE ground data processing / Archivist	LASP, Space Science Building, University of Colorado Boulder 3665 Discovery Drive Boulder, CO 80303-7813	+001 303 492 5476	Rob.Wilson@ lasp.colorado.edu
Dr Frédéric Allegrini JADE Lead and JADE Electron Instrument Scientist	Southwest Research Institute 6220 Culebra Road San Antonio, TX 78238-5166		fallegrini@swri.edu
Dr Robert W. Ebert JADE Ion Instrument Scientist	Southwest Research Institute 6220 Culebra Road San Antonio, TX 78238-5166		rebert@swri.edu
Mr John Hanley JADE Flight Software	Southwest Research Institute 6220 Culebra Road San Antonio, TX 78238-5166		jhanley@swri.edu

UCLA			
Name	**Address**	**Phone**	**Email**
Dr. Steven Joy PPI Operations Manager	IGPP, University of California 405 Hilgard Avenue Los Angeles, CA 90095-1567 USA	+001 310 825 3506	sjoy@igpp.ucla.edu
Mr. Joseph Mafi PPI Data Engineer	IGPP, University of California 405 Hilgard Avenue Los Angeles, CA 90095-1567 USA	+001 310 206 6073	jmafi@igpp.ucla.edu

JADE has had a turn-over in lead staff since launch.

- Build, pre-launch, launch to 2016-May-24:
  Dr David J. McComas was JADE Lead.
  Dr Philip Valek was the JADE Ion Instrument Scientist.
  Dr Frédéric Allegrini was the JADE Electron Instrument Scientist.
- 2016-May-24 to 2018-May-21:
  Dr Philip Valek was the JADE Lead and JADE Ion Instrument Scientist.
  Dr Frédéric Allegrini was the JADE Electron Instrument Scientist.
- 2018-May-21 onwards:
  Dr Frédéric Allegrini is the JADE Lead and JADE Electron Instrument Scientist.
  Dr Robert W. Ebert is the JADE Ion Instrument Scientist.

Appendix B PDS label files

All JADE instrument data files are accompanied by PDS label files, possessing the same names are the files they describe, but with the extension LBL. The basic content for these label files is as follows, where the NOTE field is reserved for product-specific comments:

Font below is Courier New (to equally space characters) and size 9 in order to get 78 characters to a line. This matches the PDS files that are 80 characters to a line, but the last two are \r\n.

B.1 Sample LBL file for JAD_L20_LRS_ELC_ANY_*

```
PDS_VERSION_ID = PDS3
DATA_SET_ID    = "JNO-J/SW-JAD-2-UNCALIBRATED-V1.0"

/* Input file  : JAD_L20_LRS_ELC_ANY_2015090_V01.DAT */
/* File written: 2017/05/04 23:01:39 local time  */

STANDARD_DATA_PRODUCT_ID = "JAD_L20_LRS_ELC_ANY"
PRODUCT_ID               = "JAD_L20_LRS_ELC_ANY_2015090"
PRODUCT_VERSION_ID       = "01"
PRODUCT_TYPE             = "DATA"
PRODUCT_CREATION_TIME    = 2017-125T05:01:39 /* UTC 2017-05-05 */
PROCESSING_LEVEL_ID      = "2"

RECORD_TYPE  = FIXED_LENGTH
RECORD_BYTES = 12384
FILE_RECORDS = 2

START_TIME                   = 2015-090T00:35:45.001 /* 2015-03-31 */
STOP_TIME                    = 2015-090T00:43:16.004 /* 2015-03-31 */
SPACECRAFT_CLOCK_START_COUNT = "481034275.64325" /* WHOLE.SUB (SUB 0-65535)*/
SPACECRAFT_CLOCK_STOP_COUNT  = "481034727.00000" /* Rounded nearest */
/* JADE records have start time SPACECRAFT CLOCK, so to get end time */
/* of last record, I've added the Accumulation time value to both    */
/* UTC seconds and SPACECRAFT CLOCK, - although those are not equal. */
/* Hence the SPACECRAFT_CLOCK_STOP_COUNT is rounded for now.         */

INSTRUMENT_HOST_NAME = "JUNO"
INSTRUMENT_HOST_ID   = "JNO"
TARGET_NAME          = {"JUPITER"}
INSTRUMENT_NAME      = "JOVIAN AURORAL DISTRIBUTIONS EXPERIMENT"
INSTRUMENT_ID        = "JAD" /* JADE */

DESCRIPTION = "This is the required LBL file to accompany DAT files of the
               data product JAD_L20_LRS_ELC_ANY."

MD5_CHECKSUM = "44e5efb1590fd55882dae9c00123d699"

NOTE = "See the PDS JADE SIS Document for more details on the formats."

^TABLE = "JAD_L20_LRS_ELC_ANY_2015090_V01.DAT"
OBJECT = TABLE
  INTERCHANGE_FORMAT = "BINARY"
  ROWS        = 2
  COLUMNS     = 32
  ROW_BYTES   = 12384
  DESCRIPTION = "Describes the structure and content of the data file."

/* FMT file contents start here.                                       */
/* Filename: Version01/JAD_L20_LRS_ELC_ANY_V01.FMT                      */
```

```
/* File written: 2017/05/02 15:42:00                                     */
/* Will code useful Python based letters to describe each object         */
/* see http://docs.python.org/library/struct.html for codes              */
/* formats will comma separated beginning with "RJW," as key then        */
/* {NAME}, {FORMAT}, {Number of dims}, {Size Dim 1}, {Size Dim 2}, ...   */
/* where {FORMAT} is the Python code for the type, i.e. I for uint32     */
/* and there are as many Size Dim's as number of dimensions.             */
/* Remember to remove the comment markers at either end                  */

/* RJW, BYTES_PER_RECORD, 12384 */
/* RJW, OBJECTS_PER_RECORD, 32 */

OBJECT              = COLUMN
  NAME              = SYNC
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 1
  BYTES             = 4
  VALID_MINIMUM     = 4210242563
  VALID_MAXIMUM     = 4210242563
  MISSING_CONSTANT  = 0      /* If no Sync pattern there is no record */
  DESCRIPTION       = "JADE Sync Pattern for IDP packets.
                       Hex value = 0xFAF33403, Decimal = 4210242563"
/* RJW, SYNC, I, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = DPID_COUNT
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 5
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 255
  DESCRIPTION       = "DPID Count (Source Sequence Count)
                       Count of the number of times this product has been
                       generated since the startup (or reset) of the
                       generating application (Boot Program or Science
                       Program).  This count resets to 0 upon entry to
                       the modes of BOOT, LVENG, HVENG, LOW_RATE_SCI,
                       MCP_CAL_SCI, HI_RATE_SCI, LOW_RATE_SCI2,
                       MCP_CAL_SCI2, HI_RATE_SCI2.
                       Note: starts with 0, increments by 1, eventually
                       rolls over at 255."
/* RJW, DPID_COUNT, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = COMPRESSION
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 6
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 1
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Lossless Compression Status.
                       Indicates whether the data (non-header) segment of
                       the IDP packet (IDP Data) was lossless compressed.
                         0 = Not Compressed
                         1 = Compressed"
/* RJW, COMPRESSION, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = IDPLENGTH
```

```
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 7
  BYTES              = 2
  VALID_MINIMUM      = 60 /* Depends on onboard compression, if any. */
  VALID_MAXIMUM      = 3128
  MISSING_CONSTANT   = 65535
  DESCRIPTION        = "IDP Length,
                        Byte Length of the IDP packet.
                        Uncompressed size for this product should be 3128."
/* RJW, IDPLENGTH, H, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = PACKETID
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 9
  BYTES              = 1
  VALID_MINIMUM      =   1 /* 0x01 - Range covers all JADE packets, */
  VALID_MAXIMUM      = 163 /* 0xA3 - Even those not in the PDS.     */
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Packet ID (DPID), Data Product Identifier
                        Low Rate Science - Electron Histogram
                        Only one Sensor per packet: Either E060, E180 or E300.
                        Each packet is one of the following:
                          E060 only, PACKETID = 104 /* 0x68 */
                          E180 only, PACKETID = 106 /* 0x6A */
                          E300 only, PACKETID = 107 /* 0x6B */"
/* RJW, PACKETID, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = FLIGHT_OR_STL
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 10
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 2
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "In Flight data, or STL (ground EM tests):
                            0 = In flight, from JADE on Juno (via FEI)
                            1 = On ground, from STL tests (via FEI)
                            2 = On ground, from SwRI tests (not FEI)
                          255 = Unknown"
/* RJW, FLIGHT_OR_STL, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = PACKET_MODE
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 11
  BYTES              = 1
  VALID_MINIMUM      = -2
  VALID_MAXIMUM      =  2
  MISSING_CONSTANT   = 127
  DESCRIPTION        = "Packet Mode, describes type of data telemetry.
                           -2 = HSK / Housekeeping Engineering
                           -1 = HVE / High Voltage Engineering
                            0 = CAL / MCP Calibration Science
                            1 = LRS / Low Rate Science
                            2 = HRS / High Rate Science
                          127 = Unknown
                          254 = Wrong - but HSK, see below.
                          255 = Wrong - but HVE, see below.
```

```
                       (Note, this could also be calculated via PACKETID.)
                         If you have 254 or 255 then your code is incorrect,
                       check you read a signed byte, rather than unsigned."
/* RJW, PACKET_MODE, b, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = PACKET_SPECIES
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 12
  BYTES              = 1
  VALID_MINIMUM      = -1
  VALID_MAXIMUM      =  9
  MISSING_CONSTANT   = 127
  DESCRIPTION        = "Packet Species, describes type of plasma data.
                           -1 = electrons
                            0 = ion species 0, SP0
                            1 = ion species 1, SP1
                            2 = ion species 2, SP2
                            3 = ion species 3, SP3
                            4 = ion species 4, SP4
                            5 = ion species 5, SP5
                            6 = ion species 6, SP6
                            7 = ion species 7, SP7
                            8 = Not Used
                            9 = All ions
                          127 = Unknown
                          255 = Wrong - but electrons, see below.
                          If you have 255 then your code is incorrect,
                       check you read a signed byte, rather than unsigned."
/* RJW, PACKET_SPECIES, b, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = TIMESTAMP_WHOLE
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 13
  BYTES              = 4
  VALID_MINIMUM      = 365774402 /* 2011-Aug-05: Juno Launch */
  VALID_MAXIMUM      = 599573000 /* ~ 2019-Jan-01            */
  MISSING_CONSTANT   = 0
  DESCRIPTION        = "Timestamp (Whole Second).
                       Timestamp (whole second) of the data for this packet
                       when collection began.
                       This is sometimes referred to as Mission Elapsed Time
                       (MET) and is Referenced from 2000-001T12:00:00.000 UTC,
                       but 1 tick is not exactly 1 S.I. second.
                       See UTC object for corrected converted time.
                       Note: Spacecraft Clock = TIMESTAMP_WHOLE:TIMESTAMP_SUB"
/* RJW, TIMESTAMP_WHOLE, I, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = TIMESTAMP_SUB
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 17
  BYTES              = 2
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 65535
  MISSING_CONSTANT   = 0
  DESCRIPTION        = "Timestamp (Subsecond).
                       Timestamp subsecond of the data for this packet
                       when collection began.
```

```
                       Unit: Microseconds scaled to 16 bits.
                       Note: Spacecraft Clock = TIMESTAMP_WHOLE:TIMESTAMP_SUB"
/* RJW, TIMESTAMP_SUB, H, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = ACCUMULATION_TIME
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 19
  BYTES              = 2
  VALID_MINIMUM      = 1
  VALID_MAXIMUM      = 1800
  MISSING_CONSTANT   = 65535
  UNIT               = "SECONDS"
  DESCRIPTION        = "Accumulation Time.
                       Number of seconds over which the data in this product
                       was collected (Science Program)."
/* RJW, ACCUMULATION_TIME, H, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = TABLES_VERSION
  DATA_TYPE          = PC_REAL
  START_BYTE         = 21
  BYTES              = 4
  VALID_MINIMUM      =   0.00
  VALID_MAXIMUM      =  99.99
  MISSING_CONSTANT   = -99.99
  DESCRIPTION        = "Look Up Tables (LUT) version used onboard.
                       All tables are combined (compression, sweeping,
                       macros, etc.) onboard in to a large image.
                       This is the image number, or table version.
                       Number should be to 2 decimal places."
/* RJW, TABLES_VERSION, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = FSW_VERSION
  DATA_TYPE          = PC_REAL
  START_BYTE         = 25
  BYTES              = 4
  VALID_MINIMUM      =   0.00
  VALID_MAXIMUM      =   9.99
  MISSING_CONSTANT   = -99.99
  DESCRIPTION        = "Flight Software version used.
                       Number should be to 2 decimal places."
/* RJW, FSW_VERSION, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = ACCUM_TRUNCATION
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 29
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 1
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Accumulation Truncation,
                       Whether commanded accumulation time ended early.
                           0 = Nominal
                           1 = Early
                         255 = Unknown"
/* RJW, ACCUM_TRUNCATION, B, 1, 1 */
```

```
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = DATA_UNITS /* Science Data Units only, not HSK */
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 30
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 1
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Science Data could be total counts (per accumulation)
                       or a rate, normalized to counts per view.
                         0 = All counts in the accumulation period (int)
                         1 = All counts divided by number of views (float)
                         2 = Counts per second (float)
                       255 = Not appropriate for this dataset, or Unknown."
/* RJW, DATA_UNITS, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = COMPRESSION_RATIO
  DATA_TYPE         = PC_REAL /* i.e. a float in little endian format */
  START_BYTE        = 31
  BYTES             = 4
  VALID_MINIMUM     =  1
  VALID_MAXIMUM     = 10
  MISSING_CONSTANT  = -1
  DESCRIPTION       = "Data compression ratio of data blob when it was
                       transmitted to Earth:
                           Ratio = {Uncompressed size}/{Compressed size}
                       This is the compression due to the lossless
                       scheme, and does not include any lossy compression
                       which may have occurred prior to it, such as the
                       32-bit to 8-bit or 16-bit to 8-bit look up tables
                       that are often used prior to the lossy compression.
                       A value of 1 means there was no lossless data
                       compression, i.e. it was turned off, and object
                       COMPRESSION should equal 1."
/* RJW, COMPRESSION_RATIO, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = UTC
  DATA_TYPE         = DATE  /* ASCII character string */
  START_BYTE        = 35
  BYTES             = 21
  VALID_MINIMUM     = 2011-217T00:00:00.001
                      /* SC Clock 365774402:0, JUNO Launch */
  VALID_MAXIMUM     = 2026-001T00:00:00.000 /* ~extended mission end */
  MISSING_CONSTANT  = 0001-001T00:00:00.000
  DESCRIPTION       = "UTC timestamp, of format yyyy-dddTHH:MM:SS.sss
                       where yyyy = year, ddd = day of year,
                       HH = hour, MM = minute,
                       SS.sss = decimal seconds to millisecond resolution.
                       Value calculated via SPICE from spacecraft clock time,
                       {TIMESTAMP_WHOLE}:{TIMESTAMP_SUB}"
/* RJW, UTC, c, 1, 21 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = LEAP_SECOND_VERSION
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 56
```

```
  BYTES              = 1
  VALID_MINIMUM      =  1
  VALID_MAXIMUM      = 22 /* Unpredictable, but no more than 2 a year */
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "The NAIF SPICE kernel for lsk used to generate UTC.
                        The lsk (leap second kernels) files are used in time
                        conversions and have filenames naifnnnn.tls, where
                        is the lsk version number (with leading zeros)."
/* RJW, LEAP_SECOND_VERSION, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SCLKSCET_VERSION
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 57
  BYTES              = 2
  VALID_MINIMUM      = -32767
  VALID_MAXIMUM      =  32767
  MISSING_CONSTANT   = -32768
  DESCRIPTION        = "The NAIF SPICE kernel for sclk used to generate UTC.
                        The JUNO sclk files are used to convert the spacecraft
                        clock timestamps to UTC time, and all have filenames
                        JNO_SCLKSCET.nnnnn.tsc, where nnnnn is the SCLKSCET
                        version number (with leading zeros and positive).
                        Each kernel has a reconstructed and predicted part for
                        it's values, typically any time after the last row of
                        the SCLK01_COEFFICIENTS_61999 table is predicted.
                        If TIMESTAMP_WHOLE:TIMESTAMP_SUB is in the predicted
                        part then SCLKSCET_VERSION will be negative, the
                        absolute value would be the version number.  If a later
                        SCLKSCET kernel version is used the UTC time will
                        likely be different.
                        If TIMESTAMP_WHOLE:TIMESTAMP_SUB is in the
                        reconstructed region the number will be positive
                        (equal to the version number) and will not vary with
                        later kernels.
                        e.g. If SCLKSCET_VERSION = -17 then kernel
                        JNO_SCLKSCET.00017.tsc was used to convert to UTC, but
                        it's a predicted UTC time.
                        If SCLKSCET_VERSION = 18 then kernel
                        JNO_SCLKSCET.00018.tsc was used to convert to UTC, and
                        it's a reconstructed UTC time that will not change with
                        later SCLKSCET kernel versions.
                        Within the PDS archive this value should always be
                        positive."
/* RJW, SCLKSCET_VERSION, h, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = ISSUES
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 59
  BYTES              = 4
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 4294967294
  MISSING_CONSTANT   = 4294967295
  DESCRIPTION        = "Issues or potential issues in this data record.
                       These are issues that can be identified within the JADE
                       packet of data itself without any external information.
                       e.g. timing issues due to the MAG time stutter, or any
                       voltage pulsing, would not be included as there are no
                       indicators to them within this JADE packet.
                         [For a more comprehensive list of potential issues
```

```
from internal and external sources please see the
Level 3 data.]

Level 2 issues of this JADE packet are flagged by
individual bits, and several may be hit.  If no issues
are flagged then this 4-byte unsigned integer is zero.
A value of 4294967295 is the MISSING_CONSTANT and means
that the issue status is currently unknown.

All bits at 0 implies all is okay as seen by this
packet.  If a bit is set to 1 then that bit is flagged,
otherwise it is set to zero and unflagged.

The bits are set as followed, grouped in to seriousness:

 Not very serious issues for doing science:
 Bit  0 = UTC time is predicted, yet to be finalized.
 Bit  1 = Position/Orientation values predicted, yet to
          be finalized.  Level 3 (and above) data only.
 Bit  2 = TABLES_VERSION object was altered on the
          ground to accurately reflect a 'commanded
          parameter update' outside the initial
          per-orbit commands JADE is returning.
           [If changed, the original downlinked
           TABLES_VERSION value can be found by cross-
           referencing the PARAM_TABLE_VER object in the
           JAD_L20_HSK_ALL_SHK files.  Note here the
           PARAM_TABLE_VER value is given as a unsigned
           integer of Hex Major-Middle-Minor, such that
           a value of 770 decimal is in hex 0x302,
           meaning Table Version 3.02 ]
 Bit  3 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
          fixed on the ground by adjusting
          TIMESTAMP_WHOLE, TIMESTAMP_SUB, and
          ACCUMULATION_TIME based on cross-referencing
          JADE commanding.
 Bit  4 = LRS/CAL Ion Species record with unobserved
          look directions (views) populated using views
          from neighboring record. See Bit 12 for
          uncorrected/unpopulated description.
          (Only possible if ACCUMULATION_TIME = 30.)
 Bit  5 = TIMESTAMP_WHOLE/SUB adjusted on the ground
          to mitigate any Juno time stutter affects.
          [Other TIMESTAMPs are susceptible to the
          onboard time stutter too, but only the JADE
          packet TIMESTAMP_WHOLE/SUB is tracked here.]
 Bit  6 =       Currently unused.
 Bit  7 = Warning, a leap second occurs during the
          accumulation period.

 Data slightly different than expected, but can be used
 for science with a little extra coding:
 Bit  8 = ACCUM_TRUNCATION object flagged.
 Bit  9 = Electron (HRS/LRS/CAL) MAG objects are not
          tracked, are either zeros or MISSING_CONSTANT.
            [LRS and CAL did not have MAG objects prior
             to FSW_VERSION 4.10, therefore those MAG
             objects here are set to MISSING_CONSTANT
             when FSW_VERSION < 4.10.]
 Bit 10 = TIMESTAMP_WHOLE/SUB affected by a Juno
          onboard time stutter, JADE reported timestamp
          is likely 1 whole tick too large.
          [Other TIMESTAMPs are susceptible to the
```

```
         onboard time stutter too, but only the JADE
         packet TIMESTAMP_WHOLE/SUB is tracked here.]
Bit 11 =       Currently unused.
Bit 12 = LRS/CAL Ion Species record potentially has
         unobserved look directions (spin phase sectors
         or views) present in the data, meaning the
         record may not contain data for a full 4pi
         steradians field-of-view.
         Unobserved look directions have zero counts
         per view (or counts per second) in the data,
         although an observed look direction may also
         have zero counts if no ions were measured.
         Therefore there is a potential confusion over
         zero measured counts or simply unmeasured.
         e.g. if the spin period is 30.7 seconds, then
         not all of the 78 spin phase sectors will be
         sampled in 30 seconds. (Unobserved views are
         only possible if ACCUMULATION_TIME <= 30.)
         See the JADE SIS for more information.
Bit 13 = At least one anode is blanked.
         See SIS document for further information.
Bit 14 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
         warning:
           Not fixed as yet - when fixed it will
         become bit 3 of ISSUES instead.
           Level 2 data only when FSW_VERSION = 4.00,
         ACCUMULTION_TIME object is MISSING_CONSTANT.
         Also, TIMESTAMP_WHOLE:TIMESTAMP_SUB is the end
         of the packet rather than the usual start,
         see TIMESTAMP_WHOLE object for more details.
         [Only affects data from 2015-089 to 2015-115.]
Bit 15 = Electron Anodes Reversed.
           Level 2 data only when FSW_VERSION < 4.10
         and only electron packets. Electron anodes
         are reversed in order and need to be
         remapped, however electron Spin Phase data
         (LRS data) cannot be remapped. See the SIS
         document for more information about this.
         [Affects all electron data 2011 to 2015-115.]

Data very different than expected, may not be suitable
for science - use with extreme caution.
Bit 16 = Data is not from flight instrument on Juno,
           see FLIGHT_OR_STL object.
Bit 17 = MCP_NOT_AT_COMMANDED object flagged.
           Electron HRS/CAL/HVE packets use all three
         electron sensors and therefore have three
         MCP_NOT_AT_COMMANDED values per packet.
         Setting this flag means at least one of those
         three mcps is not at its commanded value.
Bit 18 = Data includes some JADE-E300 sensor data.
         (Only flagged for HRS, LRS, CAL and HVE data.)
           E300 has a high voltage power supply issue
         and reported energy steps may be incorrect.
         If E300 is off but still reported in the data
         product, it may be zeros of fill values.
Bit 19 = Ion packet abruptly truncated.
           This packet should not be used.  It had an
         ACCUMULATION_TIME = 1, ACCUM_TRUNCATION = 1
         and the DATA object is all zeros, with a
         timestamp that matches an earlier valid packet
         that was not truncated and has non-zero DATA.
         e.g. TOF and LOG example in level 2 data at
```

```
                            TIMESTAMP_WHOLE of 495879710 (UTC 2015-261).
                   Bit 20 =       Currently unused.
                   Bit 21 =       Currently unused.
                   Bit 22 =       Currently unused.
                   Bit 23 =       Currently unused.

                   Bit 24 =       Currently unused.
                   Bit 25 =       Currently unused.
                   Bit 26 =       Currently unused.
                   Bit 27 =       Currently unused.
                   Bit 28 =       Currently unused.
                   Bit 29 =       Currently unused.
                   Bit 30 =       Currently unused.
                   Bit 31 = Reserved for MISSING_CONSTANT use.

                   Each bit has a decimal value of 2^{bit number}, and the
                   Issues flag is the sum of 2^{flagged bit numbers}.
                   For instance, if this ISSUES flag = 131329, then in
                   binary that value is 00000000000000100000000100000001
                   showing bits 17, 8 and 0 are flagged.

                   [If a currently unused bit is set, please check the
                   latest LBL file for this product that you can find to
                   see if it now has a definition.]"

    OBJECT             = BIT_COLUMN
      NAME             = ISSUES_BITS
      BIT_DATA_TYPE    = BOOLEAN
      START_BIT        =  1
      BITS             = 32
      ITEMS            = 32
      ITEM_BITS        =  1
      MINIMUM          =  0
      MAXIMUM          =  1
      DESCRIPTION      = "See ISSUES column object for description of bits."
    END_OBJECT         = BIT_COLUMN
/* RJW, ISSUES, I, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = MIN_SUBTRACTED_VALUE
  DATA_TYPE          = PC_REAL /* i.e. a float in little endian format */
  START_BYTE         = 63
  BYTES              = 4
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 65535
  MISSING_CONSTANT   = 4294967295
  UNIT               = "COUNTS/VIEW"
  DESCRIPTION        = "Minimum Subtracted Value.
                   Minimum value subtracted from every element in
                   the array data blob for transmission to Earth.
                   (This has already been added back to the DATA.)
                   Note: the units are rates (counts per views),
                   are floats rather than integers, and are fractions
                   of 1/512."
/* RJW, MIN_SUBTRACTED_VALUE, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = MCP_NOT_AT_COMMANDED
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 67
  ITEMS              = 1
```

```
  ITEM_BYTES        = 1
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 1
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "MCP not at Commanded flag.  This denotes whether
                       the MCP voltage was reduced during the data collection.
                           0 = Nominal
                           1 = Reduced
                         255 = Unknown"
/* RJW, MCP_NOT_AT_COMMANDED, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SWEEP_TABLE
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 68
  ITEMS             = 1
  ITEM_BYTES        = 1
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 2
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Which sweep table does the electron sensor have, 0-2.
                       (See PACKETID or ESENSOR object for which sensor.)"
/* RJW, SWEEP_TABLE, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = MCP_COMMANDED_VALUE
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 69
  ITEMS             = 1
  ITEM_BYTES        = 2
  BYTES             = 2
  VALID_MINIMUM     =     0
  VALID_MAXIMUM     =  4095 /* 12-bits */
  MISSING_CONSTANT  = 65535
  DESCRIPTION       = "Electron MCP Commanded raw DAC value.
                       (See PACKETID or ESENSOR object for which sensor.)"
/* RJW, MCP_COMMANDED_VALUE, H, 1, 1 */
END_OBJECT          = COLUMN

/* The following object could be treated as a 1-dimensional column of data, */
/* however we will treat it using PDS containers that allows for 1-, 2-, or */
/* 3-dimensional data. The *.DAT file is the same for both, so we give the  */
/* 1-dimension non-container description here (in comments) in case it is   */
/* useful to others for comparison.                                          */
/*                                                                           */
/*OBJECT              = COLUMN                                               */
/*  NAME              = DATA                                                 */
/*  DATA_TYPE         = PC_REAL                                              */
                      /* i.e. a float in little endian format */
/*  START_BYTE        = 71                                                   */
/*  ITEMS             = 3072                                                 */
/*  ITEM_BYTES        = 4                                                    */
/*  BYTES             = 12288                                                */
/*  VALID_MINIMUM     = 0                                                    */
/*  VALID_MAXIMUM     = 65535                                                */
/*  MISSING_CONSTANT  = 4294967295                                           */
/*  UNIT              = "COUNTS/VIEW"                                        */
/*  DESCRIPTION       = "DATA: Counts                                        */
/*                        64 Energies x 48 Electron Spin Phase Sectors.      */
```

```
/*                       The formula for mapping anodes into spin-phase     */
/*                       sectors is described in full in the PDS JADE SIS   */
/*                       and simplifies to:                                 */
/*                                                                          */
/*                   SP_sector = ( (s_phase + s_id)/7.5 + 7 - a_id ) MOD 48 */
/*                                                                          */
/*                       where:                                             */
/*                         SP_sector (electron spin phase sector) is 0 to   */
/*                           47 (rounded down to an integer).               */
/*                         s_phase is spin phase, 0 to 360 degrees.         */
/*                         s_id is the sensor in question, either 60, 180,  */
/*                           or 300.                                        */
/*                         a_id is one of the 16 anodes of the given sensor,*/
/*                           0-15. (This is anode, not the look direction.) */
/*                         7.5 degrees is the width of one anode.           */
/*                                                                          */
/*                       Note the data units are rates (counts per views),  */
/*                       are floats rather than integers, and are fractions */
/*                       of 1/512.                                          */
/*                       Note 2: Rate is independent of accumulation time.  */
/*                       Note 3: If the data is from FSW 4.00 (April 2015   */
/*                       only, when anodes were reversed - see ISSUES       */
/*                       object) then the SP_sector calculation was done    */
/*                       incorrectly.  If you must use this FSW4.00 data,   */
/*                       sum over electron spin phase sector to reduce the  */
/*                       data to energy by time and use that."              */
/* Could be, DATA, f, 1, 3072                                               */
/*END_OBJECT          = COLUMN                                              */
/*                                                                          */
/* Now follows the 2-dimensional data version using containers:             */

OBJECT               = CONTAINER
  NAME               = DATA_DIM1
  START_BYTE         = 71
  BYTES              = 192 /* = 48 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "DATA_DIM1, 2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DATA_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 48
    DESCRIPTION        = "DATA_DIM2, 1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = DATA
      DATA_TYPE          = PC_REAL /* i.e. a float in little endian format */
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      VALID_MINIMUM      = 0
      VALID_MAXIMUM      = 65535
      MISSING_CONSTANT   = 4294967295
      UNIT               = "COUNTS/VIEW"
      DESCRIPTION        = "DATA: Counts
                           64 Energies x 48 Electron Spin Phase Sectors.
                           The formula for mapping anodes into spin-phase
                           sectors is described in full in the PDS JADE SIS
                           and simplifies to:

                       SP_sector = ( (s_phase + s_id)/7.5 + 7 - a_id ) MOD 48
```

```
                          where:
                            SP_sector (electron spin phase sector) is 0 to
                              47 (rounded down to an integer).
                            s_phase is spin phase, 0 to 360 degrees.
                            s_id is the sensor in question, either 60, 180,
                              or 300.
                            a_id is one of the 16 anodes of the given sensor,
                              0-15. (This is anode, not the look direction.)
                            7.5 degrees is the width of one anode.

                          Note the data units are rates (counts per views),
                          are floats rather than integers, and are fractions
                          of 1/512.
                          Note 2: Rate is independent of accumulation time.
                          Note 3: If the data is from FSW 4.00 (April 2015
                          only, when anodes were reversed - see ISSUES
                          object) then the SP_sector calculation was done
                          incorrectly.  If you must use this FSW4.00 data,
                          sum over electron spin phase sector to reduce the
                          data to energy by time and use that."
/* RJW, DATA, f, 2, 64, 48 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = COLUMN
  NAME              = MAG_TIMESTAMP_WHOLE
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 12359
  BYTES             = 4
  VALID_MINIMUM     = 365774402 /* 2011-Aug-05: Juno Launch */
  VALID_MAXIMUM     = 599573000 /* ~ 2019-Jan-01            */
  MISSING_CONSTANT  = 0
  DESCRIPTION       = "MAG_TIMESTAMP_WHOLE
                        Whole-second timestamp of last received MAG vector
                        *before* data collection start.
                        Referenced from 12:00UTC 2000/01/01.
                        [May be affected by a Juno Time Stutter.]"
/* RJW, MAG_TIMESTAMP_WHOLE, I, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = MAG_TIMESTAMP_SUB
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 12363
  BYTES             = 2
  VALID_MINIMUM     =     0
  VALID_MAXIMUM     = 65535
  MISSING_CONSTANT  = 65535
  DESCRIPTION       = "MAG_TIMESTAMP_SUB
                        Sub-second timestamp of last received MAG vector
                        *before* data collection start.
                        A value of 65535 could be real or a MISSING_CONSTANT,
                        however it is MISSING_CONSTANT only if
                        MAG_TIMESTAMP_WHOLE = 0, e.g. WHOLE and SUB must both
                        be real or both be MISSING_CONSTANT.
                        Unit: Microseconds scaled to 16 bits.
                        [May be affected by a Juno Time Stutter.]"
/* RJW, MAG_TIMESTAMP_SUB, H, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
```

```
  NAME               = MAG_COUNT_VALID
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 12365
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 255
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "MAG_COUNT_VALID
                        Count of valid (above threshold and not saturated)
                        MAG vectors between start of *previous* packet and
                        start of this packet.
                        Note: This saturates at 255.  e.g. if there is a 600s
                        accumulation period, and the MAG vector is given every
                        2-seconds, then that's 300 counts.  If all are valid
                        then that 300 will be expressed as 255, however
                        MAG_COUNT_INVALID would still be zero."
/* RJW, MAG_COUNT_VALID, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = MAG_COUNT_INVALID
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 12366
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 255
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "MAG_COUNT_INVALID
                        Count of invalid (below threshold or saturated)
                        MAG vectors between start of *previous* packet and
                        start of this packet.
                        Note: This saturates at 255.  e.g. if there is a 600s
                        accumulation period, and the MAG vector is given every
                        2-seconds, then that's 300 counts.  If all are invalid
                        then that 300 will be expressed as 255, however
                        MAG_COUNT_VALID would still be zero."
/* RJW, MAG_COUNT_INVALID, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = MAG_VECTOR
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 12367
  ITEMS              = 3
  ITEM_BYTES         = 4
  BYTES              = 12
  VALID_MINIMUM      = -1600000
  VALID_MAXIMUM      =  1600000
  MISSING_CONSTANT   = 2147483647
  UNIT               = "nT"
  DESCRIPTION        = "Last received MAG vector in nT before
                        data collection start: 3 components [X, Y, Z]

                        MAG range is +/- 16 G, hence limits.
                        The coordinate system is spacecraft based, with:
                        +X is between E060 and E300, along the 0 degree mark
                              where E060 anode 0 starts
                        +Y is 90 degrees, between E060 anodes 11 and 12,
                        +Z is the spin axis.
                        Note: these are signed integers.

                        A Mag vector of [0 0 0] has four meanings:
                        [Meanings 1 and 2 require MAG_TIMESTAMP_WHOLE = 0.
```

```
                       The MAG_COUNT_VALID and MAG_COUNT_INVALID objects
                       can help distinguish meaning 1 from 2.]
                      1) JADE never received a mag vector at all.
                         (So initialized to 0s.)
                         e.g. MAG_COUNT_VALID = 0 for this record.
                      2) A 25s timeout has expired without JADE receiving a
                         MAG vector over a threshold magnitude.
                      [Meanings 3 and 4 require MAG_TIMESTAMP_WHOLE > 0]
                      3) The threshold parameter was set to 0 nT.
                         (Some early HVCO1 check-out data may have this.)
                      4) The broadcast message was corrupted and the
                         magnitude and components mismatched."
/* RJW, MAG_VECTOR, i, 1, 3 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = BACKGROUND_COUNTS
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 12379
  BYTES             = 4
  VALID_MINIMUM     =          0
  VALID_MAXIMUM     = 4294967294
  MISSING_CONSTANT  = 4294967295  /* 4-byte limit, rolls over */
  UNIT              = "COUNTS"
  DESCRIPTION       = "Background counts (NOT a background rate).
                      The background counter for this record's electron
                      sensor (see ESENSOR object to know which sensor).

                            This is a total count, not a rate.

                      This is a 16-bit counter over 64 energies over the
                      accumulation time (up to 1800 seconds), which means it
                      could roll over the 4-byte word. i.e. 4294967296 = 0
                      However this is unlikely, and even if so, should be
                      obvious from the visible background in object DATA."
/* RJW, BACKGROUND_COUNTS, I, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = ESENSOR
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 12383
  BYTES             = 2
  VALID_MINIMUM     =   060
  VALID_MAXIMUM     =   300
  MISSING_CONSTANT  = 65535
  DESCRIPTION       = "ESENSOR - which one of the three electron sensors is
                      this record for.  Values can only be 60, 180 or 300
                      for electron sensor E060, E180 or E300 respectively.
                      Note: each sensor also has a different PACKETID."
/* RJW, ESENSOR, H, 1, 1 */
END_OBJECT          = COLUMN
/* FMT file contents end here.                                             */

END_OBJECT = TABLE
END
```

B.2 Sample LBL file for JAD_L20_LRS_ION_ANY_*

```
PDS_VERSION_ID = PDS3
DATA_SET_ID    = "JNO-J/SW-JAD-2-UNCALIBRATED-V1.0"

/* Input file  : JAD_L20_LRS_ION_ANY_2015090_V01.DAT */
/* File written: 2017/05/04 23:03:56 local time  */

STANDARD_DATA_PRODUCT_ID = "JAD_L20_LRS_ION_ANY"
PRODUCT_ID               = "JAD_L20_LRS_ION_ANY_2015090"
PRODUCT_VERSION_ID       = "01"
PRODUCT_TYPE             = "DATA"
PRODUCT_CREATION_TIME    = 2017-125T05:03:56 /* UTC 2017-05-05 */
PROCESSING_LEVEL_ID      = "2"

RECORD_TYPE  = FIXED_LENGTH
RECORD_BYTES = 10054
FILE_RECORDS = 6

START_TIME                   = 2015-090T00:40:45.004 /* 2015-03-31 */
STOP_TIME                    = 2015-090T18:53:01.004 /* 2015-03-31 */
SPACECRAFT_CLOCK_START_COUNT = "481034575.64547" /* WHOLE.SUB (SUB 0-65535)*/
SPACECRAFT_CLOCK_STOP_COUNT  = "481100112.00000" /* Rounded nearest */
/* JADE records have start time SPACECRAFT CLOCK, so to get end time */
/* of last record, I've added the Accumulation time value to both    */
/* UTC seconds and SPACECRAFT CLOCK, - although those are not equal. */
/* Hence the SPACECRAFT_CLOCK_STOP_COUNT is rounded for now.         */

INSTRUMENT_HOST_NAME = "JUNO"
INSTRUMENT_HOST_ID   = "JNO"
TARGET_NAME          = {"JUPITER"}
INSTRUMENT_NAME      = "JOVIAN AURORAL DISTRIBUTIONS EXPERIMENT"
INSTRUMENT_ID        = "JAD" /* JADE */

DESCRIPTION = "This is the required LBL file to accompany DAT files of the
               data product JAD_L20_LRS_ION_ANY."

MD5_CHECKSUM = "ee29f7aab018fdbaeb3f9f13c3fe4d79"

NOTE = "See the PDS JADE SIS Document for more details on the formats."

^TABLE = "JAD_L20_LRS_ION_ANY_2015090_V01.DAT"
OBJECT = TABLE
  INTERCHANGE_FORMAT = "BINARY"
  ROWS        = 6
  COLUMNS     = 25
  ROW_BYTES   = 10054
  DESCRIPTION = "Describes the structure and content of the data file."

/* FMT file contents start here.                                         */
/* Filename: Version01/JAD_L20_LRS_ION_ANY_V01.FMT                       */
/* File written: 2017/05/02 15:42:00                                     */
/* Will code useful Python based letters to describe each object         */
/* see http://docs.python.org/library/struct.html for codes              */
/* formats will comma separated beginning with "RJW," as key then        */
/* {NAME}, {FORMAT}, {Number of dims}, {Size Dim 1}, {Size Dim 2}, ...   */
/* where {FORMAT} is the Python code for the type, i.e. I for uint32     */
/* and there are as many Size Dim's as number of dimensions.             */
/* Remember to remove the comment markers at either end                  */

/* RJW, BYTES_PER_RECORD, 10054 */
/* RJW, OBJECTS_PER_RECORD, 25 */
```

```
OBJECT               = COLUMN
  NAME               = SYNC
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 1
  BYTES              = 4
  VALID_MINIMUM      = 4210242563
  VALID_MAXIMUM      = 4210242563
  MISSING_CONSTANT   = 0      /* If no Sync pattern there is no record */
  DESCRIPTION        = "JADE Sync Pattern for IDP packets.
                        Hex value = 0xFAF33403, Decimal = 4210242563"
/* RJW, SYNC, I, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = DPID_COUNT
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 5
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 255
  DESCRIPTION        = "DPID Count (Source Sequence Count)
                        Count of the number of times this product has been
                        generated since the startup (or reset) of the
                        generating application (Boot Program or Science
                        Program).  This count resets to 0 upon entry to
                        the modes of BOOT, LVENG, HVENG, LOW_RATE_SCI,
                        MCP_CAL_SCI, HI_RATE_SCI, LOW_RATE_SCI2,
                        MCP_CAL_SCI2, HI_RATE_SCI2.
                        Note: starts with 0, increments by 1, eventually
                        rolls over at 255."
/* RJW, DPID_COUNT, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = COMPRESSION
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 6
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 1
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Lossless Compression Status.
                        Indicates whether the data (non-header) segment of
                        the IDP packet (IDP Data) was lossless compressed.
                          0 = Not Compressed
                          1 = Compressed"
/* RJW, COMPRESSION, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = IDPLENGTH
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 7
  BYTES              = 2
  VALID_MINIMUM      = 36 /* Depends on onboard compression, if any. */
  VALID_MAXIMUM      = 2528
  MISSING_CONSTANT   = 65535
  DESCRIPTION        = "IDP Length,
                        Byte Length of the IDP packet.
                        Uncompressed size for this product should be 2528."
/* RJW, IDPLENGTH, H, 1, 1 */
END_OBJECT           = COLUMN
```

```
OBJECT               = COLUMN
  NAME               = PACKETID
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 9
  BYTES              = 1
  VALID_MINIMUM      =   1 /* 0x01 - Range covers all JADE packets, */
  VALID_MAXIMUM      = 163 /* 0xA3 - Even those not in the PDS.     */
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Packet ID (DPID), Data Product Identifier
                         Low Rate Science - Ion Species Histogram
                         Each packet is one of the following ion species:
                           SP0, species 0, PACKETID =  96 /* 0x60 */
                           SP1, species 1, PACKETID =  97 /* 0x61 */
                           SP2, species 2, PACKETID =  98 /* 0x62 */
                           SP3, species 3, PACKETID =  99 /* 0x63 */
                           SP4, species 4, PACKETID = 100 /* 0x64 */
                           SP5, species 5, PACKETID = 101 /* 0x65 */
                           SP6, species 6, PACKETID = 102 /* 0x66 */
                           SP7, species 7, PACKETID = 103 /* 0x67 */"
/* RJW, PACKETID, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = FLIGHT_OR_STL
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 10
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 2
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "In Flight data, or STL (ground EM tests):
                             0 = In flight, from JADE on Juno (via FEI)
                             1 = On ground, from STL tests (via FEI)
                             2 = On ground, from SwRI tests (not FEI)
                           255 = Unknown"
/* RJW, FLIGHT_OR_STL, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = PACKET_MODE
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 11
  BYTES              = 1
  VALID_MINIMUM      = -2
  VALID_MAXIMUM      =  2
  MISSING_CONSTANT   = 127
  DESCRIPTION        = "Packet Mode, describes type of data telemetry.
                            -2 = HSK / Housekeeping Engineering
                            -1 = HVE / High Voltage Engineering
                             0 = CAL / MCP Calibration Science
                             1 = LRS / Low Rate Science
                             2 = HRS / High Rate Science
                           127 = Unknown
                           254 = Wrong - but HSK, see below.
                           255 = Wrong - but HVE, see below.
                         (Note, this could also be calculated via PACKETID.)
                           If you have 254 or 255 then your code is incorrect,
                         check you read a signed byte, rather than unsigned."
/* RJW, PACKET_MODE, b, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = PACKET_SPECIES
```

```
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 12
  BYTES              = 1
  VALID_MINIMUM      = -1
  VALID_MAXIMUM      =  9
  MISSING_CONSTANT   = 127
  DESCRIPTION        = "Packet Species, describes type of plasma data.
                           -1 = electrons
                            0 = ion species 0, SP0
                            1 = ion species 1, SP1
                            2 = ion species 2, SP2
                            3 = ion species 3, SP3
                            4 = ion species 4, SP4
                            5 = ion species 5, SP5
                            6 = ion species 6, SP6
                            7 = ion species 7, SP7
                            8 = Not Used
                            9 = All ions
                          127 = Unknown
                          255 = Wrong - but electrons, see below.
                          If you have 255 then your code is incorrect,
                        check you read a signed byte, rather than unsigned."
/* RJW, PACKET_SPECIES, b, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = TIMESTAMP_WHOLE
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 13
  BYTES              = 4
  VALID_MINIMUM      = 365774402 /* 2011-Aug-05: Juno Launch */
  VALID_MAXIMUM      = 599573000 /* ~ 2019-Jan-01            */
  MISSING_CONSTANT   = 0
  DESCRIPTION        = "Timestamp (Whole Second).
                        Timestamp (whole second) of the data for this packet
                        when collection began (for FSW 4.10 onwards).
                        For FSW 4.00 (April 2015 only) this is the time the
                        collection ended, rather than started, due to the LRS,
                        Ion Species Bug which was fixed in FSW 4.10.
                        This is sometimes referred to as Mission Elapsed Time
                        (MET) and is Referenced from 2000-001T12:00:00.000 UTC,
                        but 1 tick is not exactly 1 S.I. second.
                        See UTC object for corrected converted time.
                        Note: Spacecraft Clock = TIMESTAMP_WHOLE:TIMESTAMP_SUB"
/* RJW, TIMESTAMP_WHOLE, I, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = TIMESTAMP_SUB
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 17
  BYTES              = 2
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 65535
  MISSING_CONSTANT   = 0
  DESCRIPTION        = "Timestamp (Subsecond).
                        Timestamp subsecond of the data for this packet
                        when collection began (for FSW 4.10 onwards).
                        For FSW 4.00 (April 2015 only) this is the time the
                        collection ended, rather than started, due to the LRS,
                        Ion Species Bug which was fixed in FSW 4.10.
                        Unit: Microseconds scaled to 16 bits.
                        Note: Spacecraft Clock = TIMESTAMP_WHOLE:TIMESTAMP_SUB"
```

```
/* RJW, TIMESTAMP_SUB, H, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = ACCUMULATION_TIME
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 19
  BYTES             = 2
  VALID_MINIMUM     = 1
  VALID_MAXIMUM     = 1800
  MISSING_CONSTANT  = 65535
  UNIT              = "SECONDS"
  DESCRIPTION       = "Accumulation Time.
                       Number of seconds over which the data in this product
                       was collected (Science Program).
                       For FSW 4.00 (April 2015 only), Low Rate Science Ion
                       Species data had a bug where the value returned for
                       accumulation time was not relevant, and has been
                       replaced with a MISSING_CONSTANT value on the ground.
                       Use difference between time stamps to estimate
                       accumulation time."
/* RJW, ACCUMULATION_TIME, H, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = TABLES_VERSION
  DATA_TYPE         = PC_REAL
  START_BYTE        = 21
  BYTES             = 4
  VALID_MINIMUM     =   0.00
  VALID_MAXIMUM     =  99.99
  MISSING_CONSTANT  = -99.99
  DESCRIPTION       = "Look Up Tables (LUT) version used onboard.
                       All tables are combined (compression, sweeping,
                       macros, etc.) onboard in to a large image.
                       This is the image number, or table version.
                       Number should be to 2 decimal places."
/* RJW, TABLES_VERSION, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = FSW_VERSION
  DATA_TYPE         = PC_REAL
  START_BYTE        = 25
  BYTES             = 4
  VALID_MINIMUM     =   0.00
  VALID_MAXIMUM     =   9.99
  MISSING_CONSTANT  = -99.99
  DESCRIPTION       = "Flight Software version used.
                       Number should be to 2 decimal places."
/* RJW, FSW_VERSION, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = ACCUM_TRUNCATION
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 29
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 1
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Accumulation Truncation,
                       Whether commanded accumulation time ended early.
```

```
                         0 = Nominal
                         1 = Early
                       255 = Unknown"
/* RJW, ACCUM_TRUNCATION, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = DATA_UNITS /* Science Data Units only, not HSK */
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 30
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 1
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Science Data could be total counts (per accumulation)
                        or a rate, normalized to counts per view.
                          0 = All counts in the accumulation period (int)
                          1 = All counts divided by number of views (float)
                          2 = Counts per second (float)
                        255 = Not appropriate for this dataset, or Unknown."
/* RJW, DATA_UNITS, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = COMPRESSION_RATIO
  DATA_TYPE          = PC_REAL /* i.e. a float in little endian format */
  START_BYTE         = 31
  BYTES              = 4
  VALID_MINIMUM      =  1
  VALID_MAXIMUM      = 10
  MISSING_CONSTANT   = -1
  DESCRIPTION        = "Data compression ratio of data blob when it was
                        transmitted to Earth:
                            Ratio = {Uncompressed size}/{Compressed size}
                        This is the compression due to the lossless
                        scheme, and does not include any lossy compression
                        which may have occurred prior to it, such as the
                        32-bit to 8-bit or 16-bit to 8-bit look up tables
                        that are often used prior to the lossy compression.
                        A value of 1 means there was no lossless data
                        compression, i.e. it was turned off, and object
                        COMPRESSION should equal 1."
/* RJW, COMPRESSION_RATIO, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = UTC
  DATA_TYPE          = DATE  /* ASCII character string */
  START_BYTE         = 35
  BYTES              = 21
  VALID_MINIMUM      = 2011-217T00:00:00.001
                       /* SC Clock 365774402:0, JUNO Launch */
  VALID_MAXIMUM      = 2026-001T00:00:00.000 /* ~extended mission end */
  MISSING_CONSTANT   = 0001-001T00:00:00.000
  DESCRIPTION        = "UTC timestamp, of format yyyy-dddTHH:MM:SS.sss
                        where yyyy = year, ddd = day of year,
                        HH = hour, MM = minute,
                        SS.sss = decimal seconds to millisecond resolution.
                        Value calculated via SPICE from spacecraft clock time,
                        {TIMESTAMP_WHOLE}:{TIMESTAMP_SUB}"
/* RJW, UTC, c, 1, 21 */
END_OBJECT           = COLUMN
```

```
OBJECT               = COLUMN
  NAME               = LEAP_SECOND_VERSION
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 56
  BYTES              = 1
  VALID_MINIMUM      =  1
  VALID_MAXIMUM      = 22 /* Unpredictable, but no more than 2 a year */
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "The NAIF SPICE kernel for lsk used to generate UTC.
                        The lsk (leap second kernels) files are used in time
                        conversions and have filenames naifnnnn.tls, where
                        is the lsk version number (with leading zeros)."
/* RJW, LEAP_SECOND_VERSION, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SCLKSCET_VERSION
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 57
  BYTES              = 2
  VALID_MINIMUM      = -32767
  VALID_MAXIMUM      =  32767
  MISSING_CONSTANT   = -32768
  DESCRIPTION        = "The NAIF SPICE kernel for sclk used to generate UTC.
                        The JUNO sclk files are used to convert the spacecraft
                        clock timestamps to UTC time, and all have filenames
                        JNO_SCLKSCET.nnnnn.tsc, where nnnnn is the SCLKSCET
                        version number (with leading zeros and positive).
                        Each kernel has a reconstructed and predicted part for
                        it's values, typically any time after the last row of
                        the SCLK01_COEFFICIENTS_61999 table is predicted.
                        If TIMESTAMP_WHOLE:TIMESTAMP_SUB is in the predicted
                        part then SCLKSCET_VERSION will be negative, the
                        absolute value would be the version number.  If a later
                        SCLKSCET kernel version is used the UTC time will
                        likely be different.
                        If TIMESTAMP_WHOLE:TIMESTAMP_SUB is in the
                        reconstructed region the number will be positive
                        (equal to the version number) and will not vary with
                        later kernels.
                        e.g. If SCLKSCET_VERSION = -17 then kernel
                        JNO_SCLKSCET.00017.tsc was used to convert to UTC, but
                        it's a predicted UTC time.
                        If SCLKSCET_VERSION = 18 then kernel
                        JNO_SCLKSCET.00018.tsc was used to convert to UTC, and
                        it's a reconstructed UTC time that will not change with
                        later SCLKSCET kernel versions.
                        Within the PDS archive this value should always be
                        positive."
/* RJW, SCLKSCET_VERSION, h, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = ISSUES
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 59
  BYTES              = 4
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 4294967294
  MISSING_CONSTANT   = 4294967295
  DESCRIPTION        = "Issues or potential issues in this data record.
                       These are issues that can be identified within the JADE
                       packet of data itself without any external information.
```

```
e.g. timing issues due to the MAG time stutter, or any
voltage pulsing, would not be included as there are no
indicators to them within this JADE packet.
  [For a more comprehensive list of potential issues
from internal and external sources please see the
Level 3 data.]

Level 2 issues of this JADE packet are flagged by
individual bits, and several may be hit.  If no issues
are flagged then this 4-byte unsigned integer is zero.
A value of 4294967295 is the MISSING_CONSTANT and means
that the issue status is currently unknown.

All bits at 0 implies all is okay as seen by this
packet.  If a bit is set to 1 then that bit is flagged,
otherwise it is set to zero and unflagged.

The bits are set as followed, grouped in to seriousness:

 Not very serious issues for doing science:
 Bit  0 = UTC time is predicted, yet to be finalized.
 Bit  1 = Position/Orientation values predicted, yet to
          be finalized.  Level 3 (and above) data only.
 Bit  2 = TABLES_VERSION object was altered on the
          ground to accurately reflect a 'commanded
          parameter update' outside the initial
          per-orbit commands JADE is returning.
           [If changed, the original downlinked
           TABLES_VERSION value can be found by cross-
           referencing the PARAM_TABLE_VER object in the
           JAD_L20_HSK_ALL_SHK files.  Note here the
           PARAM_TABLE_VER value is given as a unsigned
           integer of Hex Major-Middle-Minor, such that
           a value of 770 decimal is in hex 0x302,
           meaning Table Version 3.02 ]
 Bit  3 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
          fixed on the ground by adjusting
          TIMESTAMP_WHOLE, TIMESTAMP_SUB, and
          ACCUMULATION_TIME based on cross-referencing
          JADE commanding.
 Bit  4 = LRS/CAL Ion Species record with unobserved
          look directions (views) populated using views
          from neighboring record. See Bit 12 for
          uncorrected/unpopulated description.
          (Only possible if ACCUMULATION_TIME = 30.)
 Bit  5 = TIMESTAMP_WHOLE/SUB adjusted on the ground
          to mitigate any Juno time stutter affects.
          [Other TIMESTAMPs are susceptible to the
          onboard time stutter too, but only the JADE
          packet TIMESTAMP_WHOLE/SUB is tracked here.]
 Bit  6 =       Currently unused.
 Bit  7 = Warning, a leap second occurs during the
          accumulation period.

 Data slightly different than expected, but can be used
 for science with a little extra coding:
 Bit  8 = ACCUM_TRUNCATION object flagged.
 Bit  9 = Electron (HRS/LRS/CAL) MAG objects are not
          tracked, are either zeros or MISSING_CONSTANT.
            [LRS and CAL did not have MAG objects prior
             to FSW_VERSION 4.10, therefore those MAG
             objects here are set to MISSING_CONSTANT
             when FSW_VERSION < 4.10.]
```

```
Bit 10 = TIMESTAMP_WHOLE/SUB affected by a Juno
         onboard time stutter, JADE reported timestamp
         is likely 1 whole tick too large.
         [Other TIMESTAMPs are susceptible to the
         onboard time stutter too, but only the JADE
         packet TIMESTAMP_WHOLE/SUB is tracked here.]
Bit 11 =       Currently unused.
Bit 12 = LRS/CAL Ion Species record potentially has
         unobserved look directions (spin phase sectors
         or views) present in the data, meaning the
         record may not contain data for a full 4pi
         steradians field-of-view.
         Unobserved look directions have zero counts
         per view (or counts per second) in the data,
         although an observed look direction may also
         have zero counts if no ions were measured.
         Therefore there is a potential confusion over
         zero measured counts or simply unmeasured.
         e.g. if the spin period is 30.7 seconds, then
         not all of the 78 spin phase sectors will be
         sampled in 30 seconds. (Unobserved views are
         only possible if ACCUMULATION_TIME <= 30.)
         See the JADE SIS for more information.
Bit 13 = At least one anode is blanked.
         See SIS document for further information.
Bit 14 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
         warning:
           Not fixed as yet - when fixed it will
         become bit 3 of ISSUES instead.
           Level 2 data only when FSW_VERSION = 4.00,
         ACCUMULTION_TIME object is MISSING_CONSTANT.
         Also, TIMESTAMP_WHOLE:TIMESTAMP_SUB is the end
         of the packet rather than the usual start,
         see TIMESTAMP_WHOLE object for more details.
         [Only affects data from 2015-089 to 2015-115.]
Bit 15 = Electron Anodes Reversed.
           Level 2 data only when FSW_VERSION < 4.10
         and only electron packets. Electron anodes
         are reversed in order and need to be
         remapped, however electron Spin Phase data
         (LRS data) cannot be remapped. See the SIS
         document for more information about this.
         [Affects all electron data 2011 to 2015-115.]

Data very different than expected, may not be suitable
for science - use with extreme caution.
Bit 16 = Data is not from flight instrument on Juno,
           see FLIGHT_OR_STL object.
Bit 17 = MCP_NOT_AT_COMMANDED object flagged.
           Electron HRS/CAL/HVE packets use all three
         electron sensors and therefore have three
         MCP_NOT_AT_COMMANDED values per packet.
         Setting this flag means at least one of those
         three mcps is not at its commanded value.
Bit 18 = Data includes some JADE-E300 sensor data.
         (Only flagged for HRS, LRS, CAL and HVE data.)
           E300 has a high voltage power supply issue
         and reported energy steps may be incorrect.
         If E300 is off but still reported in the data
         product, it may be zeros of fill values.
Bit 19 = Ion packet abruptly truncated.
           This packet should not be used.  It had an
         ACCUMULATION_TIME = 1, ACCUM_TRUNCATION = 1
```

```
                                 and the DATA object is all zeros, with a
                                 timestamp that matches an earlier valid packet
                                 that was not truncated and has non-zero DATA.
                                 e.g. TOF and LOG example in level 2 data at
                                 TIMESTAMP_WHOLE of 495879710 (UTC 2015-261).
                        Bit 20 =       Currently unused.
                        Bit 21 =       Currently unused.
                        Bit 22 =       Currently unused.
                        Bit 23 =       Currently unused.

                        Bit 24 =       Currently unused.
                        Bit 25 =       Currently unused.
                        Bit 26 =       Currently unused.
                        Bit 27 =       Currently unused.
                        Bit 28 =       Currently unused.
                        Bit 29 =       Currently unused.
                        Bit 30 =       Currently unused.
                        Bit 31 = Reserved for MISSING_CONSTANT use.

                        Each bit has a decimal value of 2^{bit number}, and the
                        Issues flag is the sum of 2^{flagged bit numbers}.
                        For instance, if this ISSUES flag = 131329, then in
                        binary that value is 00000000000000100000000100000001
                        showing bits 17, 8 and 0 are flagged.

                        [If a currently unused bit is set, please check the
                        latest LBL file for this product that you can find to
                        see if it now has a definition.]"

    OBJECT             = BIT_COLUMN
      NAME             = ISSUES_BITS
      BIT_DATA_TYPE    = BOOLEAN
      START_BIT        =  1
      BITS             = 32
      ITEMS            = 32
      ITEM_BITS        =  1
      MINIMUM          =  0
      MAXIMUM          =  1
      DESCRIPTION      = "See ISSUES column object for description of bits."
    END_OBJECT         = BIT_COLUMN
/* RJW, ISSUES, I, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = MIN_SUBTRACTED_VALUE
  DATA_TYPE          = PC_REAL /* i.e. a float in little endian format */
  START_BYTE         = 63
  BYTES              = 4
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 65535
  MISSING_CONSTANT   = 4294967295
  UNIT               = "COUNTS/VIEW"
  DESCRIPTION        = "Minimum Subtracted Value.
                        Minimum value subtracted from every element in
                        the array data blob for transmission to Earth.
                        (This has already been added back to the DATA.)
                        Note: the units are rates (counts per views),
                        are floats rather than integers, and are fractions
                        of 1/512."
/* RJW, MIN_SUBTRACTED_VALUE, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
```

```
  NAME               = MCP_NOT_AT_COMMANDED
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 67
  ITEMS              = 1
  ITEM_BYTES         = 1
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 1
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "MCP not at Commanded flag.  This denotes whether
                        the MCP voltage was reduced during the data collection.
                            0 = Nominal
                            1 = Reduced
                          255 = Unknown"
/* RJW, MCP_NOT_AT_COMMANDED, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SWEEP_TABLE
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 68
  ITEMS              = 1
  ITEM_BYTES         = 1
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 3
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Which sweep table does the ion sensor have, 0-3."
/* RJW, SWEEP_TABLE, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = MCP_COMMANDED_VALUE
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 69
  ITEMS              = 1
  ITEM_BYTES         = 2
  BYTES              = 2
  VALID_MINIMUM      =     0
  VALID_MAXIMUM      =  4095 /* 12-bits */
  MISSING_CONSTANT   = 65535
  DESCRIPTION        = "Ion MCPs Commanded raw DAC value."
/* RJW, MCP_COMMANDED_VALUE, H, 1, 1 */
END_OBJECT           = COLUMN

/* The following object could be treated as a 1-dimensional column of data, */
/* however we will treat it using PDS containers that allows for 1-, 2-, or */
/* 3-dimensional data. The *.DAT file is the same for both, so we give the  */
/* 1-dimension non-container description here (in comments) in case it is   */
/* useful to others for comparison.                                         */
/*                                                                          */
/*OBJECT               = COLUMN                                             */
/*  NAME               = DATA                                               */
/*  DATA_TYPE          = PC_REAL                                            */
                       /* i.e. a float in little endian format */
/*  START_BYTE         = 71                                                 */
/*  ITEMS              = 2496                                               */
/*  ITEM_BYTES         = 4                                                  */
/*  BYTES              = 9984                                               */
/*  VALID_MINIMUM      = 0                                                  */
/*  VALID_MAXIMUM      = 65535                                              */
/*  MISSING_CONSTANT   = 4294967295                                         */
/*  UNIT               = "COUNTS/VIEW"                                      */
```

```
/*  DESCRIPTION       = "DATA: Counts per view                          */
/*                      32 Energies x 78 Ion Spin Phase Sectors         */
/*                      The formula for mapping anodes into spin-phase  */
/*                      sectors is described in the PDS JADE SIS        */
/*                      and as follows:                                 */
/*                                                                      */
/*                        Each Spin Phase Sector has contributions from */
/*                      multiple spin phases, but always the same anode.*/
/*                        The spin phase is calculated from the start of*/
/*                      the record.                                     */
/*                        Only anodes 4-11 are used, 0-3 are not reported. */
/*                        There are 78 Spin Phase sectors [0-77] over the */
/*                      eight anodes and thirty 12-degree wide sectors, */
/*                      with spin phase sectors given in the following  */
/*                      table of anode by start spin phase:             */
/*                                                                      */
/*                       Ion Start  |              Ion Anode            */
/*                       Spin Phase | -----------------------------------*/
/*                       (Degrees)  |  4    5    6    7    8    9   10   11*/
/*                       -----------------------------------------------*/
/*                        195-207      0    3    9   24   39   54   69   75*/
/*                        207-219      0    3    9   24   39   54   69   75*/
/*                        219-231      0    3   10   25   40   55   69   75*/
/*                        231-243      0    3   10   25   40   55   69   75*/
/*                        243-255      0    3   11   26   41   56   70   75*/
/*                        255-267      0    3   11   26   41   56   70   75*/
/*                        267-279      0    4   12   27   42   57   70   75*/
/*                        279-291      0    4   12   27   42   57   70   75*/
/*                        291-303      0    4   13   28   43   58   70   75*/
/*                        303-315      0    4   13   28   43   58   70   75*/
/*                        315-327      1    5   14   29   44   59   71   76*/
/*                        327-339      1    5   14   29   44   59   71   76*/
/*                        339-351      1    5   15   30   45   60   71   76*/
/*                        351-003      1    5   15   30   45   60   71   76*/
/*                        003-015      1    5   16   31   46   61   72   76*/
/*                        015-027      1    5   16   31   46   61   72   76*/
/*                        027-039      1    6   17   32   47   62   72   76*/
/*                        039-051      1    6   17   32   47   62   72   76*/
/*                        051-063      1    6   18   33   48   63   72   76*/
/*                        063-075      1    6   18   33   48   63   72   76*/
/*                        075-087      2    7   19   34   49   64   73   77*/
/*                        087-099      2    7   19   34   49   64   73   77*/
/*                        099-111      2    7   20   35   50   65   73   77*/
/*                        111-123      2    7   20   35   50   65   73   77*/
/*                        123-135      2    7   21   36   51   66   74   77*/
/*                        135-147      2    7   21   36   51   66   74   77*/
/*                        147-159      2    8   22   37   52   67   74   77*/
/*                        159-171      2    8   22   37   52   67   74   77*/
/*                        171-183      2    8   23   38   53   68   74   77*/
/*                        183-195      2    8   23   38   53   68   74   77*/
/*                        ----------------------------------------------*/
/*                                                                      */
/*                      [The onboard software triggers on the spin phase of*/
/*                      the s/c +X axis, but JADE-I is 195 degrees further */
/*                      around, so the Ion Start Spin Phase starts at 195.]*/
/*                                                                      */
/*                      The meaning of each species is described in     */
/*                      the JADE instrument paper.                      */
/*                      Note the data units are rates (counts per views), */
/*                      are floats rather than integers, and are fractions */
/*                      of 1/512.                                       */
/*                      Note 2: Rate is independent of accumulation time." */
/* Could be, DATA, f, 1, 2496                                           */
```

```
/*END_OBJECT          = COLUMN                                               */
/*                                                                           */
/* Now follows the 2-dimensional data version using containers:              */

OBJECT              = CONTAINER
  NAME              = DATA_DIM1
  START_BYTE        = 71
  BYTES             = 312 /* = 78 * 4-bytes */
  REPETITIONS       = 32
  DESCRIPTION       = "DATA_DIM1, 2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = DATA_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 78
    DESCRIPTION       = "DATA_DIM2, 1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = DATA
      DATA_TYPE         = PC_REAL /* i.e. a float in little endian format */
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     = 0
      VALID_MAXIMUM     = 65535
      MISSING_CONSTANT  = 4294967295
      UNIT              = "COUNTS/VIEW"
      DESCRIPTION       = "DATA: Counts per view
                           32 Energies x 78 Ion Spin Phase Sectors
                           The formula for mapping anodes into spin-phase
                           sectors is described in the PDS JADE SIS
                           and as follows:

                             Each Spin Phase Sector has contributions from
                           multiple spin phases, but always the same anode.
                             The spin phase is calculated from the start of
                           the record.
                             Only anodes 4-11 are used, 0-3 are not reported.
                             There are 78 Spin Phase sectors [0-77] over the
                           eight anodes and thirty 12-degree wide sectors,
                           with spin phase sectors given in the following
                           table of anode by start spin phase:

                            Ion Start  |               Ion Anode
                            Spin Phase | ------------------------------------
                            (Degrees)  |  4    5    6    7    8    9   10   11
                            -------------------------------------------------
                             195-207      0    3    9   24   39   54   69   75
                             207-219      0    3    9   24   39   54   69   75
                             219-231      0    3   10   25   40   55   69   75
                             231-243      0    3   10   25   40   55   69   75
                             243-255      0    3   11   26   41   56   70   75
                             255-267      0    3   11   26   41   56   70   75
                             267-279      0    4   12   27   42   57   70   75
                             279-291      0    4   12   27   42   57   70   75
                             291-303      0    4   13   28   43   58   70   75
                             303-315      0    4   13   28   43   58   70   75
                             315-327      1    5   14   29   44   59   71   76
                             327-339      1    5   14   29   44   59   71   76
                             339-351      1    5   15   30   45   60   71   76
                             351-003      1    5   15   30   45   60   71   76
```

```
                              003-015      1    5   16   31   46   61   72   76
                              015-027      1    5   16   31   46   61   72   76
                              027-039      1    6   17   32   47   62   72   76
                              039-051      1    6   17   32   47   62   72   76
                              051-063      1    6   18   33   48   63   72   76
                              063-075      1    6   18   33   48   63   72   76
                              075-087      2    7   19   34   49   64   73   77
                              087-099      2    7   19   34   49   64   73   77
                              099-111      2    7   20   35   50   65   73   77
                              111-123      2    7   20   35   50   65   73   77
                              123-135      2    7   21   36   51   66   74   77
                              135-147      2    7   21   36   51   66   74   77
                              147-159      2    8   22   37   52   67   74   77
                              159-171      2    8   22   37   52   67   74   77
                              171-183      2    8   23   38   53   68   74   77
                              183-195      2    8   23   38   53   68   74   77
                              -------------------------------------------------

                             [The onboard software triggers on the spin phase of
                             the s/c +X axis, but JADE-I is 195 degrees further
                             around, so the Ion Start Spin Phase starts at 195.]

                             The meaning of each species is described in
                             the JADE instrument paper.
                             Note the data units are rates (counts per views),
                             are floats rather than integers, and are fractions
                             of 1/512.
                             Note 2: Rate is independent of accumulation time."
/* RJW, DATA, f, 2, 32, 78 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER
/* FMT file contents end here.                                                */

END_OBJECT = TABLE
END
```

Appendix C   Level 2 data record formats

This section describes the format of the Level 2 data files.

While Section 6.2.9 (“Level 2 data files”) cover this to some level, the real description is within the FMT files for each product, which themselves are embedded within the LBL files.

For details of the (very long) FMT files, please refer to the previous section (Appendix B) about label files, and the FMT files are quoted in full between these two lines within those examples:

```
/* FMT file contents start here.                                         */
                                  [FMT file in here]
/* FMT file contents end here.                                           */
```

Appendix D Level 3 data record formats

This section describes the format of the Level 3 data files.

While Section 6.2.10 ("Level 3 data files") cover this to some level, the real description is within the FMT files for each product. Here are two examples in full, but see the FMT files in the LABEL directory for specifics.

Font below is Courier New (to equally space characters) and size 9 in order to get 78 characters to a line. This matches the PDS files that are 80 characters to a line, but the last two are \r\n.

D.1 Sample FMT file for JAD_L30_HRS_ELC_TWO_CNT_V04.FMT

```
/* Filename: Version04/JAD_L30_HRS_ELC_TWO_CNT_V04.FMT                       */
/* File written: 2021/10/22 16:29:57                                         */
/* Will code useful Python based letters to describe each object             */
/* see http://docs.python.org/library/struct.html for codes                  */
/* formats will comma separated beginning with "RJW," as key then            */
/* {NAME}, {FORMAT}, {Number of dims}, {Size Dim 1}, {Size Dim 2}, ...       */
/* where {FORMAT} is the Python code for the type, i.e. I for uint32         */
/* and there are as many Size Dim's as number of dimensions.                 */
/* Remember to remove the comment markers at either end                      */

/* RJW, BYTES_PER_RECORD, 57670 */
/* RJW, OBJECTS_PER_RECORD, 52 */

OBJECT              = COLUMN
  NAME              = DIM0_UTC
  DATA_TYPE         = DATE  /* ASCII character string */
  START_BYTE        = 1
  BYTES             = 21
  VALID_MINIMUM     = 2011-217T00:00:00.001
                      /* SC Clock 365774402:0, JUNO Launch */
  VALID_MAXIMUM     = 2026-001T00:00:00.000 /* Expect mission end in 2025 */
  MISSING_CONSTANT  = 0001-001T00:00:00.000
  DESCRIPTION       = "UTC timestamp at center (not start) of record.
                       Format is yyyy-dddTHH:MM:SS.sss
                         where yyyy = year, ddd = day of year,
                         HH = hour, MM = minute,
                         SS.sss = decimal seconds to millisecond resolution.
                       Note: Duration of record can be found in S.I. seconds
                       by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse
                       this with the ACCUMULATION_TIME object, which is the
                       number of spacecraft clock ticks for accumulation.
                       While 1 tick is approximately 1 second, it is not
                       identical."
/* RJW, DIM0_UTC, c, 1, 21 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = PACKETID
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 22
  BYTES             = 1
  VALID_MINIMUM     = 142  /* (0x8E) */
  VALID_MAXIMUM     = 142  /* (0x8E) */
  MISSING_CONSTANT  = 255  /* Unknown, or a mix of packets */
```

```
  DESCRIPTION       = "Packet ID (DPID), Data Product Identifier
                       High Rate Science - Electron
                       Two Electron sensors per record: E060 and E180.
                       (This is the same data as for JAD_L30_HRS_ELC_ALL
                       but with E300 data removed for a smaller file.)
                       PACKETID = 142 (0x8E)"
/* RJW, PACKETID, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = DIM0_UTC_UPPER
  DATA_TYPE         = DATE  /* ASCII character string */
  START_BYTE        = 23
  BYTES             = 21
  VALID_MINIMUM     = 2011-217T00:00:00.001
  VALID_MAXIMUM     = 2026-001T00:00:00.000
  MISSING_CONSTANT  = 0001-001T00:00:00.000
  DESCRIPTION       = "0th Dimension of DATA: Time - upper limit.
                         See DIM0_UTC for description."
/* RJW, DIM0_UTC_UPPER, c, 1, 21 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = PACKET_MODE
  DATA_TYPE         = LSB_INTEGER
  START_BYTE        = 44
  BYTES             = 1
  VALID_MINIMUM     = 2
  VALID_MAXIMUM     = 2
  MISSING_CONSTANT  = 127
  DESCRIPTION       = "Packet Mode, describes type of data telemetry.
                          -2 = HSK / Housekeeping Engineering (Level 2 only)
                          -1 = HVE / High Voltage Engineering (Level 2 only)
                           0 = CAL / MCP Calibration Science  (Level 2 only)
                           1 = LRS / Low Rate Science
                           2 = HRS / High Rate Science
                           3 = DRS / DeRived Science from LRS and/or HRS
                         127 = Unknown
                         254 = Wrong - but HSK, see below.    (Level 2 only)
                         255 = Wrong - but HVE, see below.    (Level 2 only)
                       (Note, this could also be calculated via PACKETID.)
                         If you have 254 or 255 then your code is incorrect,
                       check you read a signed byte, rather than unsigned."
/* RJW, PACKET_MODE, b, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = DIM0_UTC_LOWER
  DATA_TYPE         = DATE  /* ASCII character string */
  START_BYTE        = 45
  BYTES             = 21
  VALID_MINIMUM     = 2011-217T00:00:00.001
  VALID_MAXIMUM     = 2026-001T00:00:00.000
  MISSING_CONSTANT  = 0001-001T00:00:00.000
  DESCRIPTION       = "0th Dimension of DATA: Time - lower limit.
                         See DIM0_UTC for description."
/* RJW, DIM0_UTC_LOWER, c, 1, 21 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = PACKET_SPECIES
  DATA_TYPE         = LSB_INTEGER
  START_BYTE        = 66
```

```
  BYTES              = 1
  VALID_MINIMUM      = -1
  VALID_MAXIMUM      = -1
  MISSING_CONSTANT   = 127
  DESCRIPTION        = "Packet Species, describes type of plasma data.
                           -1 = electrons
                            0 = ion species 0, SP0
                            1 = ion species 1, SP1
                            2 = ion species 2, SP2
                            3 = ion species 3, SP3
                            4 = ion species 4, SP4
                            5 = ion species 5, SP5
                            6 = ion species 6, SP6
                            7 = ion species 7, SP7
                            8 = Sum of SP3, SP4 and SP5
                            9 = All ions /* or any ion, e.g., TOF and LOG */
                           10 = Single ion species derived from TOF data
                          127 = Unknown
                          255 = Wrong - but electrons, see below.
                          If you have 255 then your code is incorrect,
                        check you read a signed byte, rather than unsigned."
/* RJW, PACKET_SPECIES, b, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = ACCUMULATION_TIME
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 67
  BYTES              = 2
  VALID_MINIMUM      = 1
  VALID_MAXIMUM      = 1
  MISSING_CONSTANT   = 65535
  UNIT               = "SECONDS" /* Not S.I. Seconds, but SCLK ticks */
  DESCRIPTION        = "Accumulation Time.
                        Number of seconds over which the data in this product
                        was collected (Science Program).
                        Note: Duration of record can be found in S.I. seconds
                        by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse
                        this with the ACCUMULATION_TIME object, which is the
                        number of spacecraft clock ticks for accumulation.
                        While 1 tick is approximately 1 second, it is not
                        identical.
                        ACCUMULATION_TIME is left in spacecraft clock ticks to
                        both aid matching with the level 2 data and to help
                        filtering for data taken in a particular mode."
/* RJW, ACCUMULATION_TIME, H, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = DATA_UNITS
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 69
  BYTES              = 1
  VALID_MINIMUM      = 2
  VALID_MAXIMUM      = 2
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Data units correspond to:
                            0 = All counts in the accumulation period
                            1 = All counts divided by number of views
                            2 = Counts per second
                                /* S.I. science units: */
                            3 = Differential Energy Flux [1/( m^2 sr s     )]
                            4 = Differential Number Flux [1/( m^2 sr s    J)]
```

```
                           5 = Phase Space Density      [    m^-6   s^3    ]
                               /* Convenient (non-S.I.) science units: */
                           6 = Differential Energy Flux [1/(cm^2 sr s     )]
                           7 = Differential Number Flux [1/(cm^2 sr s  keV)]
                           8 = Phase Space Density      [   cm^-6   s^3    ]
                  /* As new products are developed this list will increase */
                  /* If a number is not listed,               */
                  /*   try a LBL/FMT file from a recent date. */
                         255 = Unknown."
/* RJW, DATA_UNITS, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SOURCE_BACKGROUND
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 70
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 4
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Source of Background values (see BACKGROUND object)
                       that have been removed from the DATA object.
                           0 = None: No background has been removed
                           1 = Background anode (electron sensors only)
                           2 = Background anode (JADE-I only)
                           3 = Derived from Background anode : Method 1:
                                Background coefficients are time independent.
                               See file in CALIB directory for description.
                           4 = Derived from Background anode : Method 2:
                                Background coefficients are per orbit.
                               See file in CALIB directory for description.
 /* As new background removal methods are developed this list will increase */
                         255 = Unknown."
/* RJW, SOURCE_BACKGROUND, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SPARE_ZEROS
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 71
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 0
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Spare Zeroes.  Always zero.
                        PDS3 format required a padding byte, e.g., a 4-byte
                        integer/float will always start on the 1st or 5th
                        or 9th or 13th... byte of the record."
/* RJW, SPARE_ZEROS, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SOURCE_MAG
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 72
  BYTES              = 1
  VALID_MINIMUM      =   0
  VALID_MAXIMUM      =  39
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Source of MAG data
                       Except case 0 and 1, PAYLOAD (pl) co-ordinate MAG files
                       were used at 1s (or 2s if no 1s) resolution.
                           0 = None: No MAG data in this product.
```

```
                           1 = From Juno JADE's Level 2 files.
                               (From spacecraft and therefore uncalibrated.)
                               This is independent to JADE Level 2 version
                               number as it does not change with versions.
                               [Note MAG data in JADE files may be affected
                               by the Juno time stutter.]
                          3n = Juno's MAG's Level 3 version n calibrated
                               files, e.g., 34 means version 4, so:
                            30 = From Juno MAG's Level 3 version 00 quicklook
                                 payload files.
                                 (These are temporary files not in PDS.)
                            31 = From Juno MAG's Level 3 version 01 calibrated
                                 payload files.
                            32 = From Juno MAG's Level 3 version 02 calibrated
                                 payload files.
                            Likewise, 33 to 39 being Level 3 version 3 to 9.
                         255 = Unknown.

                          If you see a number not listed above, there may be
                        later versions of MAG data - find the latest
                        available LBL file for this product and see what that
                        has listed."
/* RJW, SOURCE_MAG, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SOURCE_JADE_METAKERNEL
  DATA_TYPE         = LSB_INTEGER
  START_BYTE        = 73
  BYTES             = 2
  VALID_MINIMUM     = -32767
  VALID_MAXIMUM     =  32767
  MISSING_CONSTANT  = -32768
  DESCRIPTION       = "The JADE SPICE metakernel used to get the time,
                       position, velocity, orientation and transformation
                       objects in this file.  The metakernel lists the
                       many individual spice kernels used, which are
                       archived by NAIF and not in this PDS volume.
                       The JADE SPICE metakernel may be found in the CALIB
                       directory of this PDS volume, with filenames of:
                             JAD_L30_SPICE_METAKERNEL_nnnnn.TXT
                       where nnnnn is the SOURCE_JADE_METAKERNEL object
                       number (with leading zeros and positive).
                       If any of the kernels within the metakernel are not
                       reconstructed (but reference or predicted) for the
                       time in question, this value will be negative.
                       Within the JADE PDS archive this value should always
                       be positive."
/* RJW, SOURCE_JADE_METAKERNEL, h, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SOURCE_JADE_CALIB
  DATA_TYPE         = LSB_INTEGER
  START_BYTE        = 75
  BYTES             = 2
  VALID_MINIMUM     =      0
  VALID_MAXIMUM     =  32767
  MISSING_CONSTANT  = -32768
  DESCRIPTION       = "The JADE calibration files list used to convert the
                       engineering units of Level 2 data to the scientific
                       units in this file.  Similar to the SPICE metakernel
                       list, this lists the many individual calibration files
```

```
                       used, each of which may be found in the CALIB
                       directory on this PDS volume.

                       This list may be found in the CALIB directory of this
                       PDS volume, with filenames of:
                             JAD_L30_CALIB_LIST_nnnnn.TXT
                       where nnnnn is the SOURCE_JADE_CALIB object
                       number (with leading zeros and positive).
                       If any of the calibration files listed are not final
                       at the time in question, this value will be negative.
                       (Newer calibration files will have a higher version and
                       simply be listed in a newer SOURCE_JADE_CALIB file.)
                       Within the JADE PDS archive this value should always
                       be positive. However, a version 00 file (for team use
                       or uploaded to JSOC, not PDS) may have negative values
                       with predicted positions/orientations/transformations."
/* RJW, SOURCE_JADE_CALIB, h, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = FSW_VERSION
  DATA_TYPE          = PC_REAL
  START_BYTE         = 77
  BYTES              = 4
  VALID_MINIMUM      =   0.00
  VALID_MAXIMUM      =   9.99
  MISSING_CONSTANT   = -99.99
  DESCRIPTION        = "Flight Software version used.
                       Number should be to 2 decimal places, with rounding.
                       e.g., 4.00, 4.10, 4.20. i.e., 4.1999998 means 4.20."
/* RJW, FSW_VERSION, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = LUT_VERSION
  DATA_TYPE          = PC_REAL
  START_BYTE         = 81
  BYTES              = 4
  VALID_MINIMUM      =   0.00
  VALID_MAXIMUM      =   9.99
  MISSING_CONSTANT   = -99.99
  DESCRIPTION        = "LUT (Look Up Table) Version used on JADE.
                       Number should be to 2 decimal places, with rounding.
                       e.g., 4.00, 4.10, 4.20. i.e., 4.1999998 means 4.20."
/* RJW, LUT_VERSION, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = LUT_VERSION_SUB_LETTER
  DATA_TYPE          = CHARACTER
  START_BYTE         = 85
  BYTES              = 2
  FORMAT             = "A2"
  DESCRIPTION        = "The letter (if any) associated with the energy table
                       used at the time of this record
                           -- -> No sub letter for this LUT Version
                           -A -> Sub letter is A for this LUT Version
                           -B -> Sub letter is B for this LUT Version
                           -C -> Sub letter is C for this LUT Version
                           etc.
                       For instance, the energy table files are in the CALIB
                       directory of this PDS volume, with names like:
                       LUT_4_00_ENERGY_V01.CSV
```

```
                        (LUT_VERSION 4.00, no sub letter)
                     or
                     LUT_5_01_K_ENERGY_V01.CSV
                         (LUT_VERSION 5.01, sub letter K)."
/* RJW, LUT_VERSION_SUB_LETTER, c, 1, 2 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = LUT_SWEEP_TABLE
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 87
  BYTES             = 1
  VALID_MINIMUM     =   1
  VALID_MAXIMUM     =  23
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "The sweep tables the ion sensor used.
                       A level 2 packet will report this as 0-3,
                       However, it requires 2 packets (a ping and a pong)
                       to make a level 3 record: either 0 and 1, or 2 and 3.
                       Therefore, a value of 1 (= 01) means sweep tables
                       0 and 1 were used, while a value of 23 means sweep
                       tables 2 and 3 were used.
                       This object can only have the value of 1 or 23."
/* RJW, LUT_SWEEP_TABLE, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = FILE_VERSION
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 88
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 4
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "The version number of the file this record came from.
                       e.g., if you loaded file
                             JAD_L30_LRS_ION_ANY_CNT_2016240_V04.DAT
                       then FILE_VERSION = 4.
                       [FILE_VERSION = 0 is never in the PDS, but is used by
                       the JADE team prior to having required calibrations.]"
/* RJW, FILE_VERSION, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_R
  DATA_TYPE         = PC_REAL
  START_BYTE        = 89
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   130.000 /* Excluding Cruise to Jupiter */
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Jupiter Radii"
  DESCRIPTION       = "Juno radial distance at time DIM0_UTC, from
                       Jupiter, in units of Jupiter Radii (Rj).
                       (1 Rj = 71492.0 km)
                       [Values may be greater than VALID_MAXIMUM
                       during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_R, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_R_UPPER
  DATA_TYPE         = PC_REAL
```

```
    START_BYTE         = 93
    BYTES              = 4
    VALID_MINIMUM      =     0.000
    VALID_MAXIMUM      =   130.000 /* Excluding Cruise to Jupiter */
    MISSING_CONSTANT   = 65535.000
    UNIT               = "Jupiter Radii"
    DESCRIPTION        = "Juno radial distance at time DIM0_UTC_UPPER, from
                          Jupiter, in units of Jupiter Radii (Rj).
                          (1 Rj = 71492.0 km)
                          SC_POS_R_UPPER could be smaller or larger than
                          SC_POS_R, depending if moving inbound or outbound.
                          [Values may be greater than VALID_MAXIMUM
                          during cruise to Jupiter before primary mission.]"
  /* RJW, SC_POS_R_UPPER, f, 1, 1 */
  END_OBJECT           = COLUMN

  OBJECT               = COLUMN
    NAME               = SC_POS_R_LOWER
    DATA_TYPE          = PC_REAL
    START_BYTE         = 97
    BYTES              = 4
    VALID_MINIMUM      =     0.000
    VALID_MAXIMUM      =   130.000 /* Excluding Cruise to Jupiter */
    MISSING_CONSTANT   = 65535.000
    UNIT               = "Jupiter Radii"
    DESCRIPTION        = "Juno radial distance at time DIM0_UTC_LOWER, from
                          Jupiter, in units of Jupiter Radii (Rj).
                          (1 Rj = 71492.0 km)
                          SC_POS_R_LOWER could be smaller or larger than
                          SC_POS_R, depending if moving inbound or outbound.
                          [Values may be greater than VALID_MAXIMUM
                          during cruise to Jupiter before primary mission.]"
  /* RJW, SC_POS_R_LOWER, f, 1, 1 */
  END_OBJECT           = COLUMN

  OBJECT               = COLUMN
    NAME               = SC_POS_LAT
    DATA_TYPE          = PC_REAL
    START_BYTE         = 101
    BYTES              = 4
    VALID_MINIMUM      =   -90.000
    VALID_MAXIMUM      =    90.000
    MISSING_CONSTANT   = 65535.000
    UNIT               = "Degrees"
    DESCRIPTION        = "Juno Latitude at time DIM0_UTC, in both the
                          IAU_JUPITER and JUNO_JSS frames, in units of degrees.
                          (0 = Equatorial)
                          (JUNO_JSS is a despun version of IAU_JUPITER, hence
                          they have identical latitudes.)"
  /* RJW, SC_POS_LAT, f, 1, 1 */
  END_OBJECT           = COLUMN

  OBJECT               = COLUMN
    NAME               = SC_POS_LAT_UPPER
    DATA_TYPE          = PC_REAL
    START_BYTE         = 105
    BYTES              = 4
    VALID_MINIMUM      =   -90.000
    VALID_MAXIMUM      =    90.000
    MISSING_CONSTANT   = 65535.000
    UNIT               = "Degrees"
    DESCRIPTION        = "Juno Latitude at time DIM0_UTC_UPPER, in both the
                          IAU_JUPITER and JUNO_JSS frames, in units of degrees.
```

```
                       (0 = Equatorial)
                       SC_POS_LAT_UPPER could be smaller or larger than
                       SC_POS_LAT.
                       (JUNO_JSS is a despun version of IAU_JUPITER, hence
                       they have identical latitudes.)"
/* RJW, SC_POS_LAT_UPPER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_LAT_LOWER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 109
  BYTES              = 4
  VALID_MINIMUM      =   -90.000
  VALID_MAXIMUM      =    90.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno Latitude at time DIM0_UTC_LOWER, in both the
                       IAU_JUPITER and JUNO_JSS frames, in units of degrees.
                       (0 = Equatorial)
                       SC_POS_LAT_LOWER could be smaller or larger than
                       SC_POS_LAT.
                       (JUNO_JSS is a despun version of IAU_JUPITER, hence
                       they have identical latitudes.)"
/* RJW, SC_POS_LAT_LOWER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_LOCAL_TIME
  DATA_TYPE          = PC_REAL
  START_BYTE         = 113
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =    24.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Hours"
  DESCRIPTION        = "Juno's (jovian) Local Time at time DIM0_UTC,
                       in units of hours.
                         00 = Midnight
                         06 = Dawn
                         12 = Noon
                         18 = Dusk"
/* RJW, SC_POS_LOCAL_TIME, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_LOCAL_TIME_UPPER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 117
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =    24.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Hours"
  DESCRIPTION        = "Juno's (jovian) Local Time at time DIM0_UTC_UPPER,
                       in units of hours.
                         00 = Midnight
                         06 = Dawn
                         12 = Noon
                         18 = Dusk"
/* RJW, SC_POS_LOCAL_TIME_UPPER, f, 1, 1 */
END_OBJECT           = COLUMN
```

```
OBJECT               = COLUMN
  NAME               = SC_POS_LOCAL_TIME_LOWER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 121
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =    24.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Hours"
  DESCRIPTION        = "Juno's (jovian) Local Time at time DIM0_UTC_LOWER,
                         in units of hours.
                           00 = Midnight
                           06 = Dawn
                           12 = Noon
                           18 = Dusk"
/* RJW, SC_POS_LOCAL_TIME_LOWER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_SYSIII_ELONG
  DATA_TYPE          = PC_REAL
  START_BYTE         = 125
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's (jovian) SYSIII (East) Longitude at time
                         DIM0_UTC, in units of degrees."
/* RJW, SC_POS_SYSIII_ELONG, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_SYSIII_ELONG_UPPER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 129
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's (jovian) SYSIII (East) Longitude at time
                         DIM0_UTC_UPPER, in units of degrees."
/* RJW, SC_POS_SYSIII_ELONG_UPPER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_SYSIII_ELONG_LOWER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 133
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's (jovian) SYSIII (East) Longitude at time
                         DIM0_UTC_LOWER, in units of degrees."
/* RJW, SC_POS_SYSIII_ELONG_LOWER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_JUPITER_J2000XYZ
  DATA_TYPE          = PC_REAL
```

```
  START_BYTE        = 137
  ITEMS             = 3
  ITEM_BYTES        = 4
  BYTES             = 12
  VALID_MINIMUM     = -10008880.0 /* ~ -140 Rj */
  VALID_MAXIMUM     =  10008880.0 /* ~ +140 Rj */
  MISSING_CONSTANT  =     65535.0 /* ~ +0.917 Rj */
  UNIT              = "km"
  DESCRIPTION       = "Juno position from Jupiter in J2000 Cartesian
                       co-ordinates [x,y,z] (units km).
                       [Values may be outside of VALID_MIN/MAX range (~140Rj)
                       during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_JUPITER_J2000XYZ, f, 1, 3 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_VEL_JUPITER_J2000XYZ
  DATA_TYPE         = PC_REAL
  START_BYTE        = 149
  ITEMS             = 3
  ITEM_BYTES        = 4
  BYTES             = 12
  VALID_MINIMUM     =   -70.0
  VALID_MAXIMUM     =    70.0
  MISSING_CONSTANT  = 65535.0
  UNIT              = "km/s"
  DESCRIPTION       = "Juno Velocity with respect to Jupiter in J2000
                       Cartesian co-ordinates [Vx,Vy,Vz] (units km/s)."
/* RJW, SC_VEL_JUPITER_J2000XYZ, f, 1, 3 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_VEL_ANGULAR_J2000XYZ
  DATA_TYPE         = PC_REAL
  START_BYTE        = 161
  ITEMS             = 3
  ITEM_BYTES        = 4
  BYTES             = 12
  VALID_MINIMUM     =    -1.0 /* General limit */
  VALID_MAXIMUM     =     1.0 /* General limit */
  MISSING_CONSTANT  = 65535.0
  UNIT              = "radians/s"
  DESCRIPTION       = "Juno Angular Velocity in Cartesian co-ordinates
                       [AVx,AVy,AVz] (units radians/s).
                         (This is calculated with the SPICE ckgpav command
                         where ref=J2000.  SPICE defines it as 'This is the
                         axis about which the reference frame tied to the
                         instrument is rotating in the right-handed sense.')"
/* RJW, SC_VEL_ANGULAR_J2000XYZ, f, 1, 3 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_SPIN_PERIOD
  DATA_TYPE         = PC_REAL
  START_BYTE        = 173
  BYTES             = 4
  VALID_MINIMUM     =     0.0
  VALID_MAXIMUM     =    70.0
  MISSING_CONSTANT  = 65535.0
  UNIT              = "SECONDS"
  DESCRIPTION       = "Juno spin period (seconds).
                       This is not useful during spacecraft maneuvers."
/* RJW, SC_SPIN_PERIOD, f, 1, 1 */
```

```
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_SPIN_PHASE
  DATA_TYPE         = PC_REAL
  START_BYTE        = 177
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   360.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno's spin phase at time DIM0_UTC,
                       in units of degrees."
/* RJW, SC_SPIN_PHASE, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_SPIN_PHASE_UPPER
  DATA_TYPE         = PC_REAL
  START_BYTE        = 181
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   360.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno's spin phase at time DIM0_UTC_UPPER,
                       in units of degrees."
/* RJW, SC_SPIN_PHASE_UPPER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_SPIN_PHASE_LOWER
  DATA_TYPE         = PC_REAL
  START_BYTE        = 185
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   360.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno's spin phase at time DIM0_UTC_LOWER,
                       in units of degrees."
/* RJW, SC_SPIN_PHASE_LOWER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = CONTAINER
  NAME              = DESPUN_SC_TO_J2000_DIM1
  START_BYTE        = 189
  BYTES             = 12 /* = 3 * 4-bytes */
  REPETITIONS       = 3
  DESCRIPTION       = "DESPUN_SC_TO_J2000_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = DESPUN_SC_TO_J2000_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 3
    DESCRIPTION       = "DESPUN_SC_TO_J2000_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = DESPUN_SC_TO_J2000
      DATA_TYPE         = PC_REAL
```

```
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =    -1.0
      VALID_MAXIMUM     =     1.0
      MISSING_CONSTANT  = 65535.0
      DESCRIPTION       = "Rotation matrix from despun spacecraft
                           co-ordinates to J2000.
                           This is a 3x3 matrix, but if read in as a 1x9
                           stream then the 1D stream is [a,b,c, d,e,f, g,h,i]
                           and the 2D matrix would be [a,b,c
                                                       d,e,f
                                                       g,h,i]"
/* RJW, DESPUN_SC_TO_J2000, f, 2, 3, 3 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = J2000_TO_JSSXYZ_DIM1
  START_BYTE        = 225
  BYTES             = 12 /* = 3 * 4-bytes */
  REPETITIONS       = 3
  DESCRIPTION       = "J2000_TO_JSSXYZ_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = J2000_TO_JSSXYZ_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 3
    DESCRIPTION       = "J2000_TO_JSSXYZ_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = J2000_TO_JSSXYZ
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =    -1.0
      VALID_MAXIMUM     =     1.0
      MISSING_CONSTANT  = 65535.0
      DESCRIPTION       = "Rotation matrix from J2000 co-ordinates to JSS xyz
                           (JSS = Jupiter-De-Spun-Sun, see SIS for details).
                           This is a 3x3 matrix, but if read in as a 1x9
                           stream then the 1D stream is [a,b,c, d,e,f, g,h,i]
                           and the 2D matrix would be [a,b,c
                                                       d,e,f
                                                       g,h,i]"
/* RJW, J2000_TO_JSSXYZ, f, 2, 3, 3 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = J2000_TO_JSSRTP_DIM1
  START_BYTE        = 261
  BYTES             = 12 /* = 3 * 4-bytes */
  REPETITIONS       = 3
  DESCRIPTION       = "J2000_TO_JSSRTP_DIM1,
```

```
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = J2000_TO_JSSRTP_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 3
    DESCRIPTION       = "J2000_TO_JSSRTP_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = J2000_TO_JSSRTP
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =    -1.0
      VALID_MAXIMUM     =     1.0
      MISSING_CONSTANT  = 65535.0
      DESCRIPTION       = "Rotation matrix from J2000 co-ordinates to
                           JSS RTP, where RTP is Jupiter centered right
                           handed R-Theta-Phi.
                           (JSS = Jupiter-De-Spun-Sun, see SIS for details.)
                           This is a 3x3 matrix, but if read in as a 1x9
                           stream then the 1D stream is [a,b,c, d,e,f, g,h,i]
                           and the 2D matrix would be [a,b,c
                                                       d,e,f
                                                       g,h,i]"
/* RJW, J2000_TO_JSSRTP, f, 2, 3, 3 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = COLUMN
  NAME              = MCP_VOLTAGE
  DATA_TYPE         = PC_REAL
  START_BYTE        = 297
  ITEMS             = 2
  ITEM_BYTES        = 4
  BYTES             = 8
  VALID_MINIMUM     = -4000.000
  VALID_MAXIMUM     =  4000.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Volts"
  DESCRIPTION       = "MCP Voltages on the two electron sensors in this
                       product, E060 and E180 respectively."
/* RJW, MCP_VOLTAGE, f, 1, 2 */
END_OBJECT          = COLUMN

OBJECT             = CONTAINER
 NAME               = ISSUES_CONTAINER
 START_BYTE         = 305
 BYTES              = 4
 REPETITIONS        = 1
 DESCRIPTION        = "ISSUES_CONTAINER, size 1."
 OBJECT             = COLUMN
  NAME              = ISSUES
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 1
  ITEMS             = 1
  ITEM_BYTES        = 4
  BYTES             = 4
```

```
VALID_MINIMUM     = 0
VALID_MAXIMUM     = 4294967294
MISSING_CONSTANT  = 4294967295
DESCRIPTION       = "Issues or potential issues in this data record.
                    These are issues that can be identified within the JADE
                    packet of data itself without any external information.
                    e.g. timing issues due to the MAG time stutter, or any
                    voltage pulsing, would not be included as there are no
                    indicators to them within this JADE packet.
                      [For a more comprehensive list of potential issues
                    from internal and external sources please see the
                    Level 3 data.]

                    Level 2 issues of this JADE packet are flagged by
                    individual bits, and several may be hit.  If no issues
                    are flagged then this 4-byte unsigned integer is zero.
                    A value of 4294967295 is the MISSING_CONSTANT and means
                    that the issue status is currently unknown.

                    All bits at 0 implies all is okay as seen by this
                    packet.  If a bit is set to 1 then that bit is flagged,
                    otherwise it is set to zero and unflagged.

                    The bits are set as followed, grouped in to seriousness:

                     Not very serious issues for doing science:
                     Bit  0 = UTC time is predicted, yet to be finalized.
                     Bit  1 = Position/Orientation values predicted, yet to
                              be finalized.  Level 3 (and above) data only.
                     Bit  2 = TABLES_VERSION object was altered on the
                              ground to accurately reflect a 'commanded
                              parameter update' outside the initial
                              per-orbit commands JADE is returning.
                               [If changed, the original downlinked
                               TABLES_VERSION value can be found by cross-
                               referencing the PARAM_TABLE_VER object in the
                               JAD_L20_HSK_ALL_SHK files.  Note here the
                               PARAM_TABLE_VER value is given as a unsigned
                               integer of Hex Major-Middle-Minor, such that
                               a value of 770 decimal is in hex 0x302,
                               meaning Table Version 3.02 ]
                     Bit  3 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
                              fixed on the ground by adjusting
                              TIMESTAMP_WHOLE, TIMESTAMP_SUB, and
                              ACCUMULATION_TIME based on cross-referencing
                              JADE commanding.
                     Bit  4 = LRS/CAL Ion Species record with unobserved
                              look directions (views) populated using views
                              from neighboring record. See Bit 12 for
                              uncorrected/unpopulated description.
                              (Only possible if ACCUMULATION_TIME = 30.)
                     Bit  5 = TIMESTAMP_WHOLE/SUB adjusted on the ground
                              to mitigate any Juno time stutter affects.
                              [Other TIMESTAMPs are susceptible to the
                              onboard time stutter too, but only the JADE
                              packet TIMESTAMP_WHOLE/SUB is tracked here.]
                     Bit  6 =       Currently unused.
                     Bit  7 = Warning, a leap second occurs during the
                              accumulation period.

                     Data slightly different than expected, but can be used
                     for science with a little extra coding:
                     Bit  8 = ACCUM_TRUNCATION object flagged.
```

```
Bit  9 = Electron (HRS/LRS/CAL) MAG objects are not
         tracked, are either zeros or MISSING_CONSTANT.
           [LRS and CAL did not have MAG objects prior
            to FSW_VERSION 4.10, therefore those MAG
            objects here are set to MISSING_CONSTANT
            when FSW_VERSION < 4.10.]
Bit 10 = TIMESTAMP_WHOLE/SUB affected by a Juno
         onboard time stutter, JADE reported timestamp
         is likely 1 whole tick too large.
         [Other TIMESTAMPs are susceptible to the
         onboard time stutter too, but only the JADE
         packet TIMESTAMP_WHOLE/SUB is tracked here.]
Bit 11 =       Currently unused.
Bit 12 = LRS/CAL Ion Species record potentially has
         unobserved look directions (spin phase sectors
         or views) present in the data, meaning the
         record may not contain data for a full 4pi
         steradians field-of-view.
         Unobserved look directions have zero counts
         per view (or counts per second) in the data,
         although an observed look direction may also
         have zero counts if no ions were measured.
         Therefore there is a potential confusion over
         zero measured counts or simply unmeasured.
         e.g. if the spin period is 30.7 seconds, then
         not all of the 78 spin phase sectors will be
         sampled in 30 seconds. (Unobserved views are
         only possible if ACCUMULATION_TIME <= 30.)
         See the JADE SIS for more information.
Bit 13 = At least one anode is blanked.
         See SIS document for further information.
Bit 14 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
         warning:
           Not fixed as yet - when fixed it will
         become bit 3 of ISSUES instead.
           Level 2 data only when FSW_VERSION = 4.00,
         ACCUMULTION_TIME object is MISSING_CONSTANT.
         Also, TIMESTAMP_WHOLE:TIMESTAMP_SUB is the end
         of the packet rather than the usual start,
         see TIMESTAMP_WHOLE object for more details.
         [Only affects data from 2015-089 to 2015-115.]
Bit 15 = Electron Anodes Reversed.
           Level 2 data only when FSW_VERSION < 4.10
         and only electron packets. Electron anodes
         are reversed in order and need to be
         remapped, however electron Spin Phase data
         (LRS data) cannot be remapped. See the SIS
         document for more information about this.
         [Affects all electron data 2011 to 2015-115.]

Data very different than expected, may not be suitable
for science - use with extreme caution.
Bit 16 = Data is not from flight instrument on Juno,
           see FLIGHT_OR_STL object.
Bit 17 = MCP_NOT_AT_COMMANDED object flagged.
           Electron HRS/CAL/HVE packets use all three
         electron sensors and therefore have three
         MCP_NOT_AT_COMMANDED values per packet.
         Setting this flag means at least one of those
         three mcps is not at its commanded value.
Bit 18 = Data includes some JADE-E300 sensor data.
         (Only flagged for HRS, LRS, CAL and HVE data.)
           E300 has a high voltage power supply issue
```

```
          and reported energy steps may be incorrect.
          If E300 is off but still reported in the data
          product, it may be zeros of fill values.
Bit 19 = Ion packet abruptly truncated.
            This packet should not be used.  It had an
          ACCUMULATION_TIME = 1, ACCUM_TRUNCATION = 1
          and the DATA object is all zeros, with a
          timestamp that matches an earlier valid packet
          that was not truncated and has non-zero DATA.
          e.g. TOF and LOG example in level 2 data at
          TIMESTAMP_WHOLE of 495879710 (UTC 2015-261).
Bit 20 = MCP Dipping Triggered, in one or more sensors.
            If the sensor measures excessive counts, it
          temporarily lowers the MCP voltage to reduce
          the number of counts and protect the sensor.
          The MCP_NOT_AT_COMMANDED object is also
          flagged (Bit 17 in ISSUES) since the MCP is
          no longer at the commanded voltage.
            For HRS/CAL/HVE electrons (datasets where
          multiple sensors are on) it is possible that
          one sensor has been dipped, but the others
          are not and still providing good data.
          (First MCP dip was HRS electrons, 2017-350.)
Bit 21 = MCP Dipped sensor's DATA set to fill values.
            If MCP dipping has triggered (Bit 20 of
          ISSUES) then:  DATA and BACKGROUND objects
          (and their *_SIGMAs) have been replaced with
          MISSING_CONSTANT values.
            (Never used for Level 2 data, which has the
          counts as measured in the dipped state.)
             In addition, Bit 17 of the ISSUES object
          (i.e. MCP_NOT_AT_COMMANDED object = 1) is set
          to zero, and, if it exists, the
          MCP_NOT_AT_COMMANDED object itself is changed
          (from 1) to be 0 for the offending sensor(s).
            If the DATA object contains data from
          multiple sensors (HRS/CAL/HVE electrons) then
          only the elements of the DATA object for the
          dipped sensor are set to MISSING_CONSTANT (as
          identified by the MCP_NOT_AT_COMMANDED value
          for each sensor (prior to setting them to 0)).
            [See Bit 22 for a similar flag.]
Bit 22 = 1 or more ELC sensor DATA set to fill values.
           Affects only electron HRS/CAL/HVE products
           (i.e. products that use multiple sensors),
           and generally only when starting that mode.
          When switching to HRS/CAL/HVE from LRS, one
          JADE-E sensor is already on, and the other(s)
          have to turn on, then it takes some time for
          that sensor to reach the commanded voltage.
          For a given record, MCP_NOT_AT_COMMANDED = 0
          for one sensor but is still = 1 for others.
          That is one sensor is taking valid science
          but the other(s) are not there yet and for
          those sensors:  DATA and BACKGROUND objects
          (and their *_SIGMAs) have been replaced
          with MISSING_CONSTANT values.
            (Never used for Level 2 data, which has the
          counts as measured in the dipped state.)
             In addition, Bit 17 of the ISSUES object
          (i.e. MCP_NOT_AT_COMMANDED object = 1) is set
          to zero, and, if it exists, the
          MCP_NOT_AT_COMMANDED object itself is changed
```

```
                             (from 1) to be 0 for the offending sensor(s).
                               Only the elements of the DATA object for the
                             original MCP_NOT_AT_COMMANDED = 1 sensor(s)
                             (prior to setting them to 0) are set to
                             MISSING_CONSTANT.
                               [Bits 21 and 22 are essentially the same
                             feature caused by an mcp voltage not being at
                             the commanded value, but the reason why this
                             is the case is different.  The treatment is
                             identical for both Bit 21 and Bit 22.]
                    Bit 23 =       Currently unused.

                    Bit 24 =       Currently unused.
                    Bit 25 =       Currently unused.
                    Bit 26 =       Currently unused.
                    Bit 27 =       Currently unused.
                    Bit 28 =       Currently unused.
                    Bit 29 =       Currently unused.
                    Bit 30 =       Currently unused.
                    Bit 31 = Reserved for MISSING_CONSTANT use.

                    Each bit has a decimal value of 2^{bit number}, and the
                    Issues flag is the sum of 2^{flagged bit numbers}.
                    For instance, if this ISSUES flag = 131329, then in
                    binary that value is 00000000000000100000000100000001
                    showing bits 17, 8 and 0 are flagged.

                    [If a currently unused bit is set, please check the
                    latest LBL file for this product that you can find to
                    see if it now has a definition.]"

    OBJECT            = BIT_COLUMN
      NAME            = ISSUES_BITS
      BIT_DATA_TYPE   = BOOLEAN
      START_BIT       =  1
      BITS            = 32
      ITEMS           = 32
      ITEM_BITS       =  1
      MINIMUM         =  0
      MAXIMUM         =  1
      DESCRIPTION     = "See ISSUES column object for description of bits."
    END_OBJECT        = BIT_COLUMN
/* RJW, ISSUES, I, 1, 1 */
 END_OBJECT        = COLUMN
END_OBJECT        = CONTAINER

OBJECT              = COLUMN
  NAME              = TIMESTAMP_WHOLE
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 309
  ITEMS             = 1
  ITEM_BYTES        = 4
  BYTES             = 4
  VALID_MINIMUM     = 365774402 /* 2011-Aug-05: Juno Launch */
  VALID_MAXIMUM     = 599573000 /* ~ 2019-Jan-01            */
  MISSING_CONSTANT  = 0
  DESCRIPTION       = "Timestamp (Whole Second) of JADE Level 2 packet
                       used to make this Level 3 record."
/* RJW, TIMESTAMP_WHOLE, I, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = TIMESTAMP_SUB
```

```
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 313
  ITEMS              = 1
  ITEM_BYTES         = 2
  BYTES              = 2
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 65535
  MISSING_CONSTANT   = 0
  DESCRIPTION        = "Timestamp (Subsecond) of JADE Level 2 packet
                        used to make this Level 3 record."
/* RJW, TIMESTAMP_SUB, H, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = CONTAINER
  NAME               = DATA_DIM1
  START_BYTE         = 315
  BYTES              = 128 /* = 32 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "DATA_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DATA_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 32
    DESCRIPTION        = "DATA_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = DATA
      DATA_TYPE          = PC_REAL
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      VALID_MINIMUM      = -999998 /* if background removed, can be <0 */
      VALID_MAXIMUM      = 2250000
      MISSING_CONSTANT   = -999999
      UNIT               = "COUNTS/SECOND"
      DESCRIPTION        = "DATA: Counts/Second
                            64 Energy x 32 Look Directions.
                            "
/* RJW, DATA, f, 2, 64, 32 */
    END_OBJECT           = COLUMN
  END_OBJECT           = CONTAINER
END_OBJECT           = CONTAINER

OBJECT               = CONTAINER
  NAME               = DATA_SIGMA_DIM1
  START_BYTE         = 8507
  BYTES              = 128 /* = 32 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "DATA_SIGMA_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DATA_SIGMA_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 32
    DESCRIPTION        = "DATA_SIGMA_DIM2,
                          1D array of data, 2nd Dimension."
```

```
    OBJECT              = COLUMN
      NAME              = DATA_SIGMA
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =       0
      VALID_MAXIMUM     =  100000
      MISSING_CONSTANT  = -999999
      UNIT              = "COUNTS/SECOND"
      DESCRIPTION       = "DATA_SIGMA
                           1-sigma uncertainties on values in object DATA,
                           such that true value = DATA +/- DATA_SIGMA.
                           See DATA entry above for size information."
/* RJW, DATA_SIGMA, f, 2, 64, 32 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = BACKGROUND_DIM1
  START_BYTE        = 16699
  BYTES             = 128 /* = 32 * 4-bytes */
  REPETITIONS       = 64
  DESCRIPTION       = "BACKGROUND_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = BACKGROUND_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 32
    DESCRIPTION       = "BACKGROUND_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = BACKGROUND
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =       0
      VALID_MAXIMUM     = 2250000
      MISSING_CONSTANT  = -999999
      UNIT              = "COUNTS/SECOND"
      DESCRIPTION       = "Background value removed from DATA.
                           No further background removal is required.
                           If you wish to do your own background removal,
                           add this object to DATA then you can remove a
                           background via your own method.
                           The background values here were found from either
                           a background anode or JADE's own ground method."
/* RJW, BACKGROUND, f, 2, 64, 32 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = BACKGROUND_SIGMA_DIM1
  START_BYTE        = 24891
```

```
  BYTES              = 128 /* = 32 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "BACKGROUND_SIGMA_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = BACKGROUND_SIGMA_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 32
    DESCRIPTION        = "BACKGROUND_SIGMA_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = BACKGROUND_SIGMA
      DATA_TYPE          = PC_REAL
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      VALID_MINIMUM      =       0
      VALID_MAXIMUM      =  100000
      MISSING_CONSTANT   = -999999
      UNIT               = "COUNTS/SECOND"
      DESCRIPTION        = "BACKGROUND_SIGMA
                            1-sigma uncertainties on values in object
                            BACKGROUND, such that
                              true value = BACKGROUND +/- BACKGROUND_SIGMA.
                            See BACKGROUND entry above for size information."
/* RJW, BACKGROUND_SIGMA, f, 2, 64, 32 */
    END_OBJECT           = COLUMN
  END_OBJECT           = CONTAINER
END_OBJECT           = CONTAINER

OBJECT               = CONTAINER
  NAME               = DIM1_E_DIM1
  START_BYTE         = 33083
  BYTES              = 128 /* = 32 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "DIM1_E_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DIM1_E_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 32
    DESCRIPTION        = "DIM1_E_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = DIM1_E
      DATA_TYPE          = PC_REAL
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      VALID_MINIMUM      =     0.0
      VALID_MAXIMUM      = 99000.0 /* Rounded up to whole keV/q */
      MISSING_CONSTANT   = 99999.0
      UNIT               = "eV/q"
      DESCRIPTION        = "1st Dimension of DATA: Energy (center) in eV/q."
/* RJW, DIM1_E, f, 2, 64, 32 */
```

```
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = DIM2_ELEVATION_DIM1
  START_BYTE        = 41275
  BYTES             = 128 /* = 32 * 4-bytes */
  REPETITIONS       = 64
  DESCRIPTION       = "DIM2_ELEVATION_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = DIM2_ELEVATION_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 32
    DESCRIPTION       = "DIM2_ELEVATION_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = DIM2_ELEVATION
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =   -90.0
      VALID_MAXIMUM     =    90.0
      MISSING_CONSTANT  = 65535.0
      UNIT              = "Degrees"
      DESCRIPTION       = "2nd Dimension of DATA: Spacecraft elevation -
                           center value. Spacecraft elevation (degs) is
                           analogous to latitude on a sphere. In spacecraft
                           xyz co-ords:
                            +z is equivalent to elevation = +90 degs
                            -z is equivalent to elevation = -90 degs
                              (The communication dish is directed along +z)
                            xy-plane at z = 0 is equivalent to elevation = 0

                           Note, 2nd dimension is really look direction
                           which has an elevation and azimuth; hence two
                           objects describe this: DIM2_ELEVATION and
                           DIM2_AZIMUTH_DESPUN."
/* RJW, DIM2_ELEVATION, f, 2, 64, 32 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = DIM2_AZIMUTH_DESPUN_DIM1
  START_BYTE        = 49467
  BYTES             = 128 /* = 32 * 4-bytes */
  REPETITIONS       = 64
  DESCRIPTION       = "DIM2_AZIMUTH_DESPUN_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = DIM2_AZIMUTH_DESPUN_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 32
    DESCRIPTION       = "DIM2_AZIMUTH_DESPUN_DIM2,
```

```
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = DIM2_AZIMUTH_DESPUN
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =     0.0
      VALID_MAXIMUM     =   360.0
      MISSING_CONSTANT  = 65535.0
      UNIT              = "Degrees"
      DESCRIPTION       = "2nd Dimension of DATA: Despun S/C azimuth -
                          center value. Spacecraft azimuth (degs) is
                          analogous to longitude on a sphere. In spacecraft
                          xyz co-ords:
                            +x is equivalent to azimuth =   0 degs
                            +y is equivalent to azimuth =  90 degs
                            -x is equivalent to azimuth = 180 degs
                            -y is equivalent to azimuth = 270 degs
                            +x is equivalent to azimuth = 360 degs
                            +y is equivalent to azimuth = 450 degs
                          The 'Despun' azimuth angle varies because Juno
                          spins, where azimuth = 0 is defined as +x when
                          spin phase equals zero (e.g., despun x-z plane
                          contains the ECLIPJ2000 north).

                          The relationship between despun azimuth and spin
                          phase is simply:
                             Despun Azimuth = 360 degrees - Spin Phase

                          Note, 2nd dimension is really look direction
                          which has an elevation and azimuth; hence two
                          objects describe this: DIM2_ELEVATION and
                          DIM2_AZIMUTH_DESPUN."
/* RJW, DIM2_AZIMUTH_DESPUN, f, 2, 64, 32 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = COLUMN
  NAME              = MAG_VECTOR
  DATA_TYPE         = PC_REAL
  START_BYTE        = 57659
  ITEMS             = 3
  ITEM_BYTES        = 4
  BYTES             = 12
  VALID_MINIMUM     = -1600000.0
  VALID_MAXIMUM     =  1600000.0
  MISSING_CONSTANT  =  9990000.0
  UNIT              = "nT"
  DESCRIPTION       = "MAG vector in nT, 3 components [X, Y, Z]
                      MAG range is +/- 16 G, hence limits.
                      This xyz coordinate system is despun spacecraft; see
                      the definitions of DIM2_ELEVATION and DIM2_AZIMUTH:
                        +X is when [azimuth, elevation] = [  0,  0] degrees,
                        +Y is when [azimuth, elevation] = [ 90,  0] degrees,
                        +Z is when elevation = 90 degrees."
/* RJW, MAG_VECTOR, f, 1, 3 */
END_OBJECT          = COLUMN
```

D.2 Sample FMT file for JAD_L30_HLS_ION_TOF_CNT_V04.FMT

```
/* Filename: Version04/JAD_L30_HLS_ION_TOF_CNT_V04.FMT                       */
/* File written: 2021/10/22 16:29:57                                         */
/* Will code useful Python based letters to describe each object             */
/* see http://docs.python.org/library/struct.html for codes                  */
/* formats will comma separated beginning with "RJW," as key then            */
/* {NAME}, {FORMAT}, {Number of dims}, {Size Dim 1}, {Size Dim 2}, ...       */
/* where {FORMAT} is the Python code for the type, i.e. I for uint32         */
/* and there are as many Size Dim's as number of dimensions.                 */
/* Remember to remove the comment markers at either end                      */

/* RJW, BYTES_PER_RECORD, 98228 */
/* RJW, OBJECTS_PER_RECORD, 58 */

OBJECT             = COLUMN
  NAME             = DIM0_UTC
  DATA_TYPE        = DATE  /* ASCII character string */
  START_BYTE       = 1
  BYTES            = 21
  VALID_MINIMUM    = 2011-217T00:00:00.001
                     /* SC Clock 365774402:0, JUNO Launch */
  VALID_MAXIMUM    = 2026-001T00:00:00.000 /* Expect mission end in 2025 */
  MISSING_CONSTANT = 0001-001T00:00:00.000
  DESCRIPTION      = "UTC timestamp at center (not start) of record.
                      Format is yyyy-dddTHH:MM:SS.sss
                        where yyyy = year, ddd = day of year,
                        HH = hour, MM = minute,
                        SS.sss = decimal seconds to millisecond resolution.
                      Note: Duration of record can be found in S.I. seconds
                      by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse
                      this with the ACCUMULATION_TIME object, which is the
                      number of spacecraft clock ticks for accumulation.
                      While 1 tick is approximately 1 second, it is not
                      identical."
/* RJW, DIM0_UTC, c, 1, 21 */
END_OBJECT         = COLUMN

OBJECT             = COLUMN
  NAME             = PACKETID
  DATA_TYPE        = LSB_UNSIGNED_INTEGER
  START_BYTE       = 22
  BYTES            = 1
  VALID_MINIMUM    = 105  /* (0x69) */
  VALID_MAXIMUM    = 137  /* (0x89) */
  MISSING_CONSTANT = 255  /* Unknown, or a mix of packets */
  DESCRIPTION      = "Packet ID (DPID), Data Product Identifier
                      High and Low Rate Science - Ion Time Of Flight
                        PACKETID = 137 (0x89) = High Rate Science
                        PACKETID = 105 (0x69) =  Low Rate Science"
/* RJW, PACKETID, B, 1, 1 */
END_OBJECT         = COLUMN

OBJECT             = COLUMN
  NAME             = DIM0_UTC_UPPER
  DATA_TYPE        = DATE  /* ASCII character string */
  START_BYTE       = 23
  BYTES            = 21
  VALID_MINIMUM    = 2011-217T00:00:00.001
  VALID_MAXIMUM    = 2026-001T00:00:00.000
  MISSING_CONSTANT = 0001-001T00:00:00.000
  DESCRIPTION      = "0th Dimension of DATA: Time - upper limit.
```

```
                         See DIM0_UTC for description."
/* RJW, DIM0_UTC_UPPER, c, 1, 21 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = PACKET_MODE
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 44
  BYTES              = 1
  VALID_MINIMUM      = 1
  VALID_MAXIMUM      = 2
  MISSING_CONSTANT   = 127
  DESCRIPTION        = "Packet Mode, describes type of data telemetry.
                          -2 = HSK / Housekeeping Engineering (Level 2 only)
                          -1 = HVE / High Voltage Engineering (Level 2 only)
                           0 = CAL / MCP Calibration Science  (Level 2 only)
                           1 = LRS / Low Rate Science
                           2 = HRS / High Rate Science
                           3 = DRS / DeRived Science from LRS and/or HRS
                         127 = Unknown
                         254 = Wrong - but HSK, see below.    (Level 2 only)
                         255 = Wrong - but HVE, see below.    (Level 2 only)
                       (Note, this could also be calculated via PACKETID.)
                         If you have 254 or 255 then your code is incorrect,
                       check you read a signed byte, rather than unsigned."
/* RJW, PACKET_MODE, b, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = DIM0_UTC_LOWER
  DATA_TYPE          = DATE  /* ASCII character string */
  START_BYTE         = 45
  BYTES              = 21
  VALID_MINIMUM      = 2011-217T00:00:00.001
  VALID_MAXIMUM      = 2026-001T00:00:00.000
  MISSING_CONSTANT   = 0001-001T00:00:00.000
  DESCRIPTION        = "0th Dimension of DATA: Time - lower limit.
                         See DIM0_UTC for description."
/* RJW, DIM0_UTC_LOWER, c, 1, 21 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = PACKET_SPECIES
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 66
  BYTES              = 1
  VALID_MINIMUM      = 9
  VALID_MAXIMUM      = 9
  MISSING_CONSTANT   = 127
  DESCRIPTION        = "Packet Species, describes type of plasma data.
                          -1 = electrons
                           0 = ion species 0, SP0
                           1 = ion species 1, SP1
                           2 = ion species 2, SP2
                           3 = ion species 3, SP3
                           4 = ion species 4, SP4
                           5 = ion species 5, SP5
                           6 = ion species 6, SP6
                           7 = ion species 7, SP7
                           8 = Sum of SP3, SP4 and SP5
                           9 = All ions /* or any ion, e.g., TOF and LOG */
                          10 = Single ion species derived from TOF data
                         127 = Unknown
```

```
                       255 = Wrong - but electrons, see below.
                       If you have 255 then your code is incorrect,
                     check you read a signed byte, rather than unsigned."
/* RJW, PACKET_SPECIES, b, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = ACCUMULATION_TIME
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 67
  BYTES              = 2
  VALID_MINIMUM      = 1
  VALID_MAXIMUM      = 1800
  MISSING_CONSTANT   = 65535
  UNIT               = "SECONDS" /* Not S.I. Seconds, but SCLK ticks */
  DESCRIPTION        = "Accumulation Time.
                       Number of seconds over which the data in this product
                       was collected (Science Program).
                       Note: Duration of record can be found in S.I. seconds
                       by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse
                       this with the ACCUMULATION_TIME object, which is the
                       number of spacecraft clock ticks for accumulation.
                       While 1 tick is approximately 1 second, it is not
                       identical.
                       ACCUMULATION_TIME is left in spacecraft clock ticks to
                       both aid matching with the level 2 data and to help
                       filtering for data taken in a particular mode."
/* RJW, ACCUMULATION_TIME, H, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = DATA_UNITS
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 69
  BYTES              = 1
  VALID_MINIMUM      = 2
  VALID_MAXIMUM      = 2
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Data units correspond to:
                           0 = All counts in the accumulation period
                           1 = All counts divided by number of views
                           2 = Counts per second
                               /* S.I. science units: */
                           3 = Differential Energy Flux [1/( m^2 sr s     )]
                           4 = Differential Number Flux [1/( m^2 sr s    J)]
                           5 = Phase Space Density      [    m^-6   s^3    ]
                               /* Convenient (non-S.I.) science units: */
                           6 = Differential Energy Flux [1/(cm^2 sr s     )]
                           7 = Differential Number Flux [1/(cm^2 sr s  keV)]
                           8 = Phase Space Density      [   cm^-6   s^3    ]
                  /* As new products are developed this list will increase */
                  /* If a number is not listed,               */
                  /*   try a LBL/FMT file from a recent date. */
                       255 = Unknown."
/* RJW, DATA_UNITS, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SOURCE_BACKGROUND
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 70
  BYTES              = 1
  VALID_MINIMUM      = 0
```

```
  VALID_MAXIMUM     = 4
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Source of Background values (see BACKGROUND object)
                       that have been removed from the DATA object.
                           0 = None: No background has been removed
                           1 = Background anode (electron sensors only)
                           2 = Background anode (JADE-I only)
                           3 = Derived from Background anode : Method 1:
                                Background coefficients are time independent.
                               See file in CALIB directory for description.
                           4 = Derived from Background anode : Method 2:
                                Background coefficients are per orbit.
                               See file in CALIB directory for description.
 /* As new background removal methods are developed this list will increase */
                         255 = Unknown."
/* RJW, SOURCE_BACKGROUND, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SOURCE_SPECIES_REMAPPED
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 71
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 0
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Source of ion remapping for ION TOF data products:
                           0 = None: Data has not been remapped on the ground.
                         255 = Unknown."
/* RJW, SOURCE_SPECIES_REMAPPED, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SOURCE_MAG
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 72
  BYTES             = 1
  VALID_MINIMUM     =   0
  VALID_MAXIMUM     =  39
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Source of MAG data
                       Except case 0 and 1, PAYLOAD (pl) co-ordinate MAG files
                       were used at 1s (or 2s if no 1s) resolution.
                           0 = None: No MAG data in this product.
                           1 = From Juno JADE's Level 2 files.
                               (From spacecraft and therefore uncalibrated.)
                               This is independent to JADE Level 2 version
                               number as it does not change with versions.
                               [Note MAG data in JADE files may be affected
                               by the Juno time stutter.]
                          3n = Juno's MAG's Level 3 version n calibrated
                               files, e.g., 34 means version 4, so:
                            30 = From Juno MAG's Level 3 version 00 quicklook
                                 payload files.
                                 (These are temporary files not in PDS.)
                            31 = From Juno MAG's Level 3 version 01 calibrated
                                 payload files.
                            32 = From Juno MAG's Level 3 version 02 calibrated
                                 payload files.
                            Likewise, 33 to 39 being Level 3 version 3 to 9.
                         255 = Unknown.

                          If you see a number not listed above, there may be
```

```
                        later versions of MAG data - find the latest
                        available LBL file for this product and see what that
                        has listed."
/* RJW, SOURCE_MAG, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SOURCE_JADE_METAKERNEL
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 73
  BYTES              = 2
  VALID_MINIMUM      = -32767
  VALID_MAXIMUM      =  32767
  MISSING_CONSTANT   = -32768
  DESCRIPTION        = "The JADE SPICE metakernel used to get the time,
                        position, velocity, orientation and transformation
                        objects in this file.  The metakernel lists the
                        many individual spice kernels used, which are
                        archived by NAIF and not in this PDS volume.
                        The JADE SPICE metakernel may be found in the CALIB
                        directory of this PDS volume, with filenames of:
                              JAD_L30_SPICE_METAKERNEL_nnnnn.TXT
                        where nnnnn is the SOURCE_JADE_METAKERNEL object
                        number (with leading zeros and positive).
                        If any of the kernels within the metakernel are not
                        reconstructed (but reference or predicted) for the
                        time in question, this value will be negative.
                        Within the JADE PDS archive this value should always
                        be positive."
/* RJW, SOURCE_JADE_METAKERNEL, h, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SOURCE_JADE_CALIB
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 75
  BYTES              = 2
  VALID_MINIMUM      =      0
  VALID_MAXIMUM      =  32767
  MISSING_CONSTANT   = -32768
  DESCRIPTION        = "The JADE calibration files list used to convert the
                        engineering units of Level 2 data to the scientific
                        units in this file.  Similar to the SPICE metakernel
                        list, this lists the many individual calibration files
                        used, each of which may be found in the CALIB
                        directory on this PDS volume.

                        This list may be found in the CALIB directory of this
                        PDS volume, with filenames of:
                              JAD_L30_CALIB_LIST_nnnnn.TXT
                        where nnnnn is the SOURCE_JADE_CALIB object
                        number (with leading zeros and positive).
                        If any of the calibration files listed are not final
                        at the time in question, this value will be negative.
                        (Newer calibration files will have a higher version and
                        simply be listed in a newer SOURCE_JADE_CALIB file.)
                        Within the JADE PDS archive this value should always
                        be positive. However, a version 00 file (for team use
                        or uploaded to JSOC, not PDS) may have negative values
                        with predicted positions/orientations/transformations."
/* RJW, SOURCE_JADE_CALIB, h, 1, 1 */
END_OBJECT           = COLUMN
```

```
OBJECT               = COLUMN
  NAME               = FSW_VERSION
  DATA_TYPE          = PC_REAL
  START_BYTE         = 77
  BYTES              = 4
  VALID_MINIMUM      =   0.00
  VALID_MAXIMUM      =   9.99
  MISSING_CONSTANT   = -99.99
  DESCRIPTION        = "Flight Software version used.
                        Number should be to 2 decimal places, with rounding.
                        e.g., 4.00, 4.10, 4.20. i.e., 4.1999998 means 4.20."
/* RJW, FSW_VERSION, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = LUT_VERSION
  DATA_TYPE          = PC_REAL
  START_BYTE         = 81
  BYTES              = 4
  VALID_MINIMUM      =   0.00
  VALID_MAXIMUM      =   9.99
  MISSING_CONSTANT   = -99.99
  DESCRIPTION        = "LUT (Look Up Table) Version used on JADE.
                        Number should be to 2 decimal places, with rounding.
                        e.g., 4.00, 4.10, 4.20. i.e., 4.1999998 means 4.20."
/* RJW, LUT_VERSION, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = LUT_VERSION_SUB_LETTER
  DATA_TYPE          = CHARACTER
  START_BYTE         = 85
  BYTES              = 2
  FORMAT             = "A2"
  DESCRIPTION        = "The letter (if any) associated with the energy table
                        used at the time of this record
                            -- -> No sub letter for this LUT Version
                            -A -> Sub letter is A for this LUT Version
                            -B -> Sub letter is B for this LUT Version
                            -C -> Sub letter is C for this LUT Version
                            etc.
                        For instance, the energy table files are in the CALIB
                        directory of this PDS volume, with names like:
                        LUT_4_00_ENERGY_V01.CSV
                           (LUT_VERSION 4.00, no sub letter)
                        or
                        LUT_5_01_K_ENERGY_V01.CSV
                            (LUT_VERSION 5.01, sub letter K)."
/* RJW, LUT_VERSION_SUB_LETTER, c, 1, 2 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = LUT_SWEEP_TABLE
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 87
  BYTES              = 1
  VALID_MINIMUM      =   1
  VALID_MAXIMUM      =  23
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "The sweep tables the ion sensor used.
                        A level 2 packet will report this as 0-3,
                        However, it requires 2 packets (a ping and a pong)
                        to make a level 3 record: either 0 and 1, or 2 and 3.
```

```
                       Therefore, a value of 1 (= 01) means sweep tables
                       0 and 1 were used, while a value of 23 means sweep
                       tables 2 and 3 were used.
                       This object can only have the value of 1 or 23."
/* RJW, LUT_SWEEP_TABLE, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = FILE_VERSION
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 88
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 4
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "The version number of the file this record came from.
                       e.g., if you loaded file
                             JAD_L30_LRS_ION_ANY_CNT_2016240_V04.DAT
                       then FILE_VERSION = 4.
                       [FILE_VERSION = 0 is never in the PDS, but is used by
                       the JADE team prior to having required calibrations.]"
/* RJW, FILE_VERSION, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_R
  DATA_TYPE         = PC_REAL
  START_BYTE        = 89
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   130.000 /* Excluding Cruise to Jupiter */
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Jupiter Radii"
  DESCRIPTION       = "Juno radial distance at time DIM0_UTC, from
                       Jupiter, in units of Jupiter Radii (Rj).
                       (1 Rj = 71492.0 km)
                       [Values may be greater than VALID_MAXIMUM
                       during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_R, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_R_UPPER
  DATA_TYPE         = PC_REAL
  START_BYTE        = 93
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   130.000 /* Excluding Cruise to Jupiter */
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Jupiter Radii"
  DESCRIPTION       = "Juno radial distance at time DIM0_UTC_UPPER, from
                       Jupiter, in units of Jupiter Radii (Rj).
                       (1 Rj = 71492.0 km)
                       SC_POS_R_UPPER could be smaller or larger than
                       SC_POS_R, depending if moving inbound or outbound.
                       [Values may be greater than VALID_MAXIMUM
                       during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_R_UPPER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_R_LOWER
  DATA_TYPE         = PC_REAL
```

```
  START_BYTE         = 97
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   130.000 /* Excluding Cruise to Jupiter */
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Jupiter Radii"
  DESCRIPTION        = "Juno radial distance at time DIM0_UTC_LOWER, from
                         Jupiter, in units of Jupiter Radii (Rj).
                         (1 Rj = 71492.0 km)
                         SC_POS_R_LOWER could be smaller or larger than
                         SC_POS_R, depending if moving inbound or outbound.
                         [Values may be greater than VALID_MAXIMUM
                         during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_R_LOWER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_LAT
  DATA_TYPE          = PC_REAL
  START_BYTE         = 101
  BYTES              = 4
  VALID_MINIMUM      =   -90.000
  VALID_MAXIMUM      =    90.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno Latitude at time DIM0_UTC, in both the
                         IAU_JUPITER and JUNO_JSS frames, in units of degrees.
                         (0 = Equatorial)
                         (JUNO_JSS is a despun version of IAU_JUPITER, hence
                         they have identical latitudes.)"
/* RJW, SC_POS_LAT, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_LAT_UPPER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 105
  BYTES              = 4
  VALID_MINIMUM      =   -90.000
  VALID_MAXIMUM      =    90.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno Latitude at time DIM0_UTC_UPPER, in both the
                         IAU_JUPITER and JUNO_JSS frames, in units of degrees.
                         (0 = Equatorial)
                         SC_POS_LAT_UPPER could be smaller or larger than
                         SC_POS_LAT.
                         (JUNO_JSS is a despun version of IAU_JUPITER, hence
                         they have identical latitudes.)"
/* RJW, SC_POS_LAT_UPPER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_LAT_LOWER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 109
  BYTES              = 4
  VALID_MINIMUM      =   -90.000
  VALID_MAXIMUM      =    90.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno Latitude at time DIM0_UTC_LOWER, in both the
                         IAU_JUPITER and JUNO_JSS frames, in units of degrees.
```

```
                       (0 = Equatorial)
                       SC_POS_LAT_LOWER could be smaller or larger than
                       SC_POS_LAT.
                       (JUNO_JSS is a despun version of IAU_JUPITER, hence
                       they have identical latitudes.)"
/* RJW, SC_POS_LAT_LOWER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_LOCAL_TIME
  DATA_TYPE         = PC_REAL
  START_BYTE        = 113
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =    24.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Hours"
  DESCRIPTION       = "Juno's (jovian) Local Time at time DIM0_UTC,
                       in units of hours.
                         00 = Midnight
                         06 = Dawn
                         12 = Noon
                         18 = Dusk"
/* RJW, SC_POS_LOCAL_TIME, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_LOCAL_TIME_UPPER
  DATA_TYPE         = PC_REAL
  START_BYTE        = 117
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =    24.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Hours"
  DESCRIPTION       = "Juno's (jovian) Local Time at time DIM0_UTC_UPPER,
                       in units of hours.
                         00 = Midnight
                         06 = Dawn
                         12 = Noon
                         18 = Dusk"
/* RJW, SC_POS_LOCAL_TIME_UPPER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_LOCAL_TIME_LOWER
  DATA_TYPE         = PC_REAL
  START_BYTE        = 121
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =    24.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Hours"
  DESCRIPTION       = "Juno's (jovian) Local Time at time DIM0_UTC_LOWER,
                       in units of hours.
                         00 = Midnight
                         06 = Dawn
                         12 = Noon
                         18 = Dusk"
/* RJW, SC_POS_LOCAL_TIME_LOWER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
```

```
  NAME               = SC_POS_SYSIII_ELONG
  DATA_TYPE          = PC_REAL
  START_BYTE         = 125
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's (jovian) SYSIII (East) Longitude at time
                        DIM0_UTC, in units of degrees."
/* RJW, SC_POS_SYSIII_ELONG, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_SYSIII_ELONG_UPPER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 129
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's (jovian) SYSIII (East) Longitude at time
                        DIM0_UTC_UPPER, in units of degrees."
/* RJW, SC_POS_SYSIII_ELONG_UPPER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_SYSIII_ELONG_LOWER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 133
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's (jovian) SYSIII (East) Longitude at time
                        DIM0_UTC_LOWER, in units of degrees."
/* RJW, SC_POS_SYSIII_ELONG_LOWER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_JUPITER_J2000XYZ
  DATA_TYPE          = PC_REAL
  START_BYTE         = 137
  ITEMS              = 3
  ITEM_BYTES         = 4
  BYTES              = 12
  VALID_MINIMUM      = -10008880.0 /* ~ -140 Rj */
  VALID_MAXIMUM      =  10008880.0 /* ~ +140 Rj */
  MISSING_CONSTANT   =     65535.0 /* ~ +0.917 Rj */
  UNIT               = "km"
  DESCRIPTION        = "Juno position from Jupiter in J2000 Cartesian
                        co-ordinates [x,y,z] (units km).
                        [Values may be outside of VALID_MIN/MAX range (~140Rj)
                        during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_JUPITER_J2000XYZ, f, 1, 3 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_VEL_JUPITER_J2000XYZ
  DATA_TYPE          = PC_REAL
  START_BYTE         = 149
```

```
  ITEMS             = 3
  ITEM_BYTES        = 4
  BYTES             = 12
  VALID_MINIMUM     =   -70.0
  VALID_MAXIMUM     =    70.0
  MISSING_CONSTANT  = 65535.0
  UNIT              = "km/s"
  DESCRIPTION       = "Juno Velocity with respect to Jupiter in J2000
                       Cartesian co-ordinates [Vx,Vy,Vz] (units km/s)."
/* RJW, SC_VEL_JUPITER_J2000XYZ, f, 1, 3 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_VEL_ANGULAR_J2000XYZ
  DATA_TYPE         = PC_REAL
  START_BYTE        = 161
  ITEMS             = 3
  ITEM_BYTES        = 4
  BYTES             = 12
  VALID_MINIMUM     =    -1.0 /* General limit */
  VALID_MAXIMUM     =     1.0 /* General limit */
  MISSING_CONSTANT  = 65535.0
  UNIT              = "radians/s"
  DESCRIPTION       = "Juno Angular Velocity in Cartesian co-ordinates
                       [AVx,AVy,AVz] (units radians/s).
                         (This is calculated with the SPICE ckgpav command
                         where ref=J2000.  SPICE defines it as 'This is the
                         axis about which the reference frame tied to the
                         instrument is rotating in the right-handed sense.')"
/* RJW, SC_VEL_ANGULAR_J2000XYZ, f, 1, 3 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_SPIN_PERIOD
  DATA_TYPE         = PC_REAL
  START_BYTE        = 173
  BYTES             = 4
  VALID_MINIMUM     =     0.0
  VALID_MAXIMUM     =    70.0
  MISSING_CONSTANT  = 65535.0
  UNIT              = "SECONDS"
  DESCRIPTION       = "Juno spin period (seconds).
                       This is not useful during spacecraft maneuvers."
/* RJW, SC_SPIN_PERIOD, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_SPIN_PHASE
  DATA_TYPE         = PC_REAL
  START_BYTE        = 177
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   360.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno's spin phase at time DIM0_UTC,
                       in units of degrees."
/* RJW, SC_SPIN_PHASE, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_SPIN_PHASE_UPPER
  DATA_TYPE         = PC_REAL
```

```
  START_BYTE        = 181
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   360.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno's spin phase at time DIM0_UTC_UPPER,
                       in units of degrees."
/* RJW, SC_SPIN_PHASE_UPPER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_SPIN_PHASE_LOWER
  DATA_TYPE         = PC_REAL
  START_BYTE        = 185
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   360.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno's spin phase at time DIM0_UTC_LOWER,
                       in units of degrees."
/* RJW, SC_SPIN_PHASE_LOWER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = CONTAINER
  NAME              = DESPUN_SC_TO_J2000_DIM1
  START_BYTE        = 189
  BYTES             = 12 /* = 3 * 4-bytes */
  REPETITIONS       = 3
  DESCRIPTION       = "DESPUN_SC_TO_J2000_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = DESPUN_SC_TO_J2000_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 3
    DESCRIPTION       = "DESPUN_SC_TO_J2000_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = DESPUN_SC_TO_J2000
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =    -1.0
      VALID_MAXIMUM     =     1.0
      MISSING_CONSTANT  = 65535.0
      DESCRIPTION       = "Rotation matrix from despun spacecraft
                           co-ordinates to J2000.
                           This is a 3x3 matrix, but if read in as a 1x9
                           stream then the 1D stream is [a,b,c, d,e,f, g,h,i]
                           and the 2D matrix would be [a,b,c
                                                       d,e,f
                                                       g,h,i]"
/* RJW, DESPUN_SC_TO_J2000, f, 2, 3, 3 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER
```

```
OBJECT               = CONTAINER
  NAME               = J2000_TO_JSSXYZ_DIM1
  START_BYTE         = 225
  BYTES              = 12 /* = 3 * 4-bytes */
  REPETITIONS        = 3
  DESCRIPTION        = "J2000_TO_JSSXYZ_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = J2000_TO_JSSXYZ_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 3
    DESCRIPTION        = "J2000_TO_JSSXYZ_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = J2000_TO_JSSXYZ
      DATA_TYPE          = PC_REAL
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      VALID_MINIMUM      =    -1.0
      VALID_MAXIMUM      =     1.0
      MISSING_CONSTANT   = 65535.0
      DESCRIPTION        = "Rotation matrix from J2000 co-ordinates to JSS xyz
                            (JSS = Jupiter-De-Spun-Sun, see SIS for details).
                            This is a 3x3 matrix, but if read in as a 1x9
                            stream then the 1D stream is [a,b,c, d,e,f, g,h,i]
                            and the 2D matrix would be [a,b,c
                                                        d,e,f
                                                        g,h,i]"
/* RJW, J2000_TO_JSSXYZ, f, 2, 3, 3 */
    END_OBJECT           = COLUMN
  END_OBJECT           = CONTAINER
END_OBJECT           = CONTAINER

OBJECT               = CONTAINER
  NAME               = J2000_TO_JSSRTP_DIM1
  START_BYTE         = 261
  BYTES              = 12 /* = 3 * 4-bytes */
  REPETITIONS        = 3
  DESCRIPTION        = "J2000_TO_JSSRTP_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = J2000_TO_JSSRTP_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 3
    DESCRIPTION        = "J2000_TO_JSSRTP_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = J2000_TO_JSSRTP
      DATA_TYPE          = PC_REAL
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      VALID_MINIMUM      =    -1.0
      VALID_MAXIMUM      =     1.0
```

```
      MISSING_CONSTANT  = 65535.0
      DESCRIPTION       = "Rotation matrix from J2000 co-ordinates to
                           JSS RTP, where RTP is Jupiter centered right
                           handed R-Theta-Phi.
                           (JSS = Jupiter-De-Spun-Sun, see SIS for details.)
                           This is a 3x3 matrix, but if read in as a 1x9
                           stream then the 1D stream is [a,b,c, d,e,f, g,h,i]
                           and the 2D matrix would be [a,b,c
                                                        d,e,f
                                                        g,h,i]"
/* RJW, J2000_TO_JSSRTP, f, 2, 3, 3 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = COLUMN
  NAME              = MCP_VOLTAGE
  DATA_TYPE         = PC_REAL
  START_BYTE        = 297
  ITEMS             = 1
  ITEM_BYTES        = 4
  BYTES             = 4
  VALID_MINIMUM     = -4000.000
  VALID_MAXIMUM     =  4000.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Volts"
  DESCRIPTION       = "MCP Voltage on sensor."
/* RJW, MCP_VOLTAGE, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT            = CONTAINER
 NAME              = ISSUES_CONTAINER
 START_BYTE        = 301
 BYTES             = 4
 REPETITIONS       = 2
 DESCRIPTION       = "ISSUES_CONTAINER, size 2."
 OBJECT            = COLUMN
  NAME              = ISSUES
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 1
  ITEMS             = 1
  ITEM_BYTES        = 4
  BYTES             = 4
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 4294967294
  MISSING_CONSTANT  = 4294967295
  DESCRIPTION       = "Issues or potential issues in this data record.
                         [Two values for ions as this is the ISSUES object
                          from both the ping and pong level 2 packets used
                          to create this record.]
                      These are issues that can be identified within the JADE
                      packet of data itself without any external information.
                      e.g. timing issues due to the MAG time stutter, or any
                      voltage pulsing, would not be included as there are no
                      indicators to them within this JADE packet.
                         [For a more comprehensive list of potential issues
                      from internal and external sources please see the
                      Level 3 data.]

                      Level 2 issues of this JADE packet are flagged by
                      individual bits, and several may be hit.  If no issues
                      are flagged then this 4-byte unsigned integer is zero.
                      A value of 4294967295 is the MISSING_CONSTANT and means
```

that the issue status is currently unknown.

All bits at 0 implies all is okay as seen by this packet. If a bit is set to 1 then that bit is flagged, otherwise it is set to zero and unflagged.

The bits are set as followed, grouped in to seriousness:

```
 Not very serious issues for doing science:
 Bit  0 = UTC time is predicted, yet to be finalized.
 Bit  1 = Position/Orientation values predicted, yet to
          be finalized.  Level 3 (and above) data only.
 Bit  2 = TABLES_VERSION object was altered on the
          ground to accurately reflect a 'commanded
          parameter update' outside the initial
          per-orbit commands JADE is returning.
           [If changed, the original downlinked
           TABLES_VERSION value can be found by cross-
           referencing the PARAM_TABLE_VER object in the
           JAD_L20_HSK_ALL_SHK files.  Note here the
           PARAM_TABLE_VER value is given as a unsigned
           integer of Hex Major-Middle-Minor, such that
           a value of 770 decimal is in hex 0x302,
           meaning Table Version 3.02 ]
 Bit  3 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
          fixed on the ground by adjusting
          TIMESTAMP_WHOLE, TIMESTAMP_SUB, and
          ACCUMULATION_TIME based on cross-referencing
          JADE commanding.
 Bit  4 = LRS/CAL Ion Species record with unobserved
          look directions (views) populated using views
          from neighboring record. See Bit 12 for
          uncorrected/unpopulated description.
          (Only possible if ACCUMULATION_TIME = 30.)
 Bit  5 = TIMESTAMP_WHOLE/SUB adjusted on the ground
          to mitigate any Juno time stutter affects.
          [Other TIMESTAMPs are susceptible to the
          onboard time stutter too, but only the JADE
          packet TIMESTAMP_WHOLE/SUB is tracked here.]
 Bit  6 =       Currently unused.
 Bit  7 = Warning, a leap second occurs during the
          accumulation period.

 Data slightly different than expected, but can be used
 for science with a little extra coding:
 Bit  8 = ACCUM_TRUNCATION object flagged.
 Bit  9 = Electron (HRS/LRS/CAL) MAG objects are not
          tracked, are either zeros or MISSING_CONSTANT.
            [LRS and CAL did not have MAG objects prior
             to FSW_VERSION 4.10, therefore those MAG
             objects here are set to MISSING_CONSTANT
             when FSW_VERSION < 4.10.]
 Bit 10 = TIMESTAMP_WHOLE/SUB affected by a Juno
          onboard time stutter, JADE reported timestamp
          is likely 1 whole tick too large.
          [Other TIMESTAMPs are susceptible to the
          onboard time stutter too, but only the JADE
          packet TIMESTAMP_WHOLE/SUB is tracked here.]
 Bit 11 =       Currently unused.
 Bit 12 = LRS/CAL Ion Species record potentially has
          unobserved look directions (spin phase sectors
          or views) present in the data, meaning the
          record may not contain data for a full 4pi
```

```
            steradians field-of-view.
            Unobserved look directions have zero counts
            per view (or counts per second) in the data,
            although an observed look direction may also
            have zero counts if no ions were measured.
            Therefore there is a potential confusion over
            zero measured counts or simply unmeasured.
            e.g. if the spin period is 30.7 seconds, then
            not all of the 78 spin phase sectors will be
            sampled in 30 seconds. (Unobserved views are
            only possible if ACCUMULATION_TIME <= 30.)
            See the JADE SIS for more information.
   Bit 13 = At least one anode is blanked.
            See SIS document for further information.
   Bit 14 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
            warning:
              Not fixed as yet - when fixed it will
            become bit 3 of ISSUES instead.
              Level 2 data only when FSW_VERSION = 4.00,
            ACCUMULTION_TIME object is MISSING_CONSTANT.
            Also, TIMESTAMP_WHOLE:TIMESTAMP_SUB is the end
            of the packet rather than the usual start,
            see TIMESTAMP_WHOLE object for more details.
            [Only affects data from 2015-089 to 2015-115.]
   Bit 15 = Electron Anodes Reversed.
              Level 2 data only when FSW_VERSION < 4.10
            and only electron packets. Electron anodes
            are reversed in order and need to be
            remapped, however electron Spin Phase data
            (LRS data) cannot be remapped. See the SIS
            document for more information about this.
            [Affects all electron data 2011 to 2015-115.]

   Data very different than expected, may not be suitable
   for science - use with extreme caution.
   Bit 16 = Data is not from flight instrument on Juno,
              see FLIGHT_OR_STL object.
   Bit 17 = MCP_NOT_AT_COMMANDED object flagged.
              Electron HRS/CAL/HVE packets use all three
            electron sensors and therefore have three
            MCP_NOT_AT_COMMANDED values per packet.
            Setting this flag means at least one of those
            three mcps is not at its commanded value.
   Bit 18 = Data includes some JADE-E300 sensor data.
            (Only flagged for HRS, LRS, CAL and HVE data.)
              E300 has a high voltage power supply issue
            and reported energy steps may be incorrect.
            If E300 is off but still reported in the data
            product, it may be zeros of fill values.
   Bit 19 = Ion packet abruptly truncated.
              This packet should not be used.  It had an
            ACCUMULATION_TIME = 1, ACCUM_TRUNCATION = 1
            and the DATA object is all zeros, with a
            timestamp that matches an earlier valid packet
            that was not truncated and has non-zero DATA.
            e.g. TOF and LOG example in level 2 data at
            TIMESTAMP_WHOLE of 495879710 (UTC 2015-261).
   Bit 20 = MCP Dipping Triggered, in one or more sensors.
              If the sensor measures excessive counts, it
            temporarily lowers the MCP voltage to reduce
            the number of counts and protect the sensor.
            The MCP_NOT_AT_COMMANDED object is also
            flagged (Bit 17 in ISSUES) since the MCP is
```

```
          no longer at the commanded voltage.
            For HRS/CAL/HVE electrons (datasets where
          multiple sensors are on) it is possible that
          one sensor has been dipped, but the others
          are not and still providing good data.
          (First MCP dip was HRS electrons, 2017-350.)
Bit 21 = MCP Dipped sensor's DATA set to fill values.
            If MCP dipping has triggered (Bit 20 of
          ISSUES) then:  DATA and BACKGROUND objects
          (and their *_SIGMAs) have been replaced with
          MISSING_CONSTANT values.
            (Never used for Level 2 data, which has the
          counts as measured in the dipped state.)
             In addition, Bit 17 of the ISSUES object
          (i.e. MCP_NOT_AT_COMMANDED object = 1) is set
          to zero, and, if it exists, the
          MCP_NOT_AT_COMMANDED object itself is changed
          (from 1) to be 0 for the offending sensor(s).
            If the DATA object contains data from
          multiple sensors (HRS/CAL/HVE electrons) then
          only the elements of the DATA object for the
          dipped sensor are set to MISSING_CONSTANT (as
          identified by the MCP_NOT_AT_COMMANDED value
          for each sensor (prior to setting them to 0)).
            [See Bit 22 for a similar flag.]
Bit 22 = 1 or more ELC sensor DATA set to fill values.
           Affects only electron HRS/CAL/HVE products
           (i.e. products that use multiple sensors),
           and generally only when starting that mode.
          When switching to HRS/CAL/HVE from LRS, one
          JADE-E sensor is already on, and the other(s)
          have to turn on, then it takes some time for
          that sensor to reach the commanded voltage.
          For a given record, MCP_NOT_AT_COMMANDED = 0
          for one sensor but is still = 1 for others.
          That is one sensor is taking valid science
          but the other(s) are not there yet and for
          those sensors:  DATA and BACKGROUND objects
          (and their *_SIGMAs) have been replaced
          with MISSING_CONSTANT values.
            (Never used for Level 2 data, which has the
          counts as measured in the dipped state.)
             In addition, Bit 17 of the ISSUES object
          (i.e. MCP_NOT_AT_COMMANDED object = 1) is set
          to zero, and, if it exists, the
          MCP_NOT_AT_COMMANDED object itself is changed
          (from 1) to be 0 for the offending sensor(s).
            Only the elements of the DATA object for the
          original MCP_NOT_AT_COMMANDED = 1 sensor(s)
          (prior to setting them to 0) are set to
          MISSING_CONSTANT.
            [Bits 21 and 22 are essentially the same
          feature caused by an mcp voltage not being at
          the commanded value, but the reason why this
          is the case is different.  The treatment is
          identical for both Bit 21 and Bit 22.]
Bit 23 =       Currently unused.

Bit 24 =       Currently unused.
Bit 25 =       Currently unused.
Bit 26 =       Currently unused.
Bit 27 =       Currently unused.
Bit 28 =       Currently unused.
```

```
                        Bit 29 =       Currently unused.
                        Bit 30 =       Currently unused.
                        Bit 31 = Reserved for MISSING_CONSTANT use.

                        Each bit has a decimal value of 2^{bit number}, and the
                        Issues flag is the sum of 2^{flagged bit numbers}.
                        For instance, if this ISSUES flag = 131329, then in
                        binary that value is 00000000000000100000000100000001
                        showing bits 17, 8 and 0 are flagged.

                        [If a currently unused bit is set, please check the
                        latest LBL file for this product that you can find to
                        see if it now has a definition.]"

    OBJECT             = BIT_COLUMN
      NAME             = ISSUES_BITS
      BIT_DATA_TYPE    = BOOLEAN
      START_BIT        =  1
      BITS             = 32
      ITEMS            = 32
      ITEM_BITS        =  1
      MINIMUM          =  0
      MAXIMUM          =  1
      DESCRIPTION      = "See ISSUES column object for description of bits."
    END_OBJECT         = BIT_COLUMN
/* RJW, ISSUES, I, 1, 2 */
 END_OBJECT        = COLUMN
END_OBJECT        = CONTAINER

OBJECT               = COLUMN
  NAME               = TIMESTAMP_WHOLE
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 309
  ITEMS              = 2
  ITEM_BYTES         = 4
  BYTES              = 8
  VALID_MINIMUM      = 365774402 /* 2011-Aug-05: Juno Launch */
  VALID_MAXIMUM      = 599573000 /* ~ 2019-Jan-01            */
  MISSING_CONSTANT   = 0
  DESCRIPTION        = "Timestamps (Whole Second) of JADE Level 2 packets
                        used to make this Level 3 record.
                        (Both the ping and pong level 2 packets.)"
/* RJW, TIMESTAMP_WHOLE, I, 1, 2 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = TIMESTAMP_SUB
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 317
  ITEMS              = 2
  ITEM_BYTES         = 2
  BYTES              = 4
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 65535
  MISSING_CONSTANT   = 0
  DESCRIPTION        = "Timestamps (Subsecond) of JADE Level 2 packets
                        used to make this Level 3 record.
                        (Both the ping and pong level 2 packets.)"
/* RJW, TIMESTAMP_SUB, H, 1, 2 */
END_OBJECT           = COLUMN

OBJECT               = CONTAINER
  NAME               = DATA_DIM1
```

```
  START_BYTE         = 321
  BYTES              = 372 /* = 1 * 93 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "DATA_DIM1,
                        3D array of data, 1st, 2nd and 3rd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DATA_DIM2
    START_BYTE         = 1
    BYTES              = 372 /* = 93 * 4-bytes */
    REPETITIONS        = 1
    DESCRIPTION        = "DATA_DIM2,
                          2D array of data, 2nd Dimension."

    OBJECT               = CONTAINER
      NAME               = DATA_DIM3
      START_BYTE         = 1
      BYTES              = 4
      REPETITIONS        = 93
      DESCRIPTION        = "DATA_DIM3,
                          1D array of data, 3rd Dimension."

      OBJECT               = COLUMN
        NAME               = DATA
        DATA_TYPE          = PC_REAL
        START_BYTE         = 1
        ITEMS              = 1
        ITEM_BYTES         = 4
        BYTES              = 4
        VALID_MINIMUM      = -999998 /* if background removed, can be <0 */
        VALID_MAXIMUM      = 2250000
        MISSING_CONSTANT   = -999999
        UNIT               = "COUNTS/SECOND"
        DESCRIPTION        = "DATA: Counts/Second
                              64 Energy x 1 Look Direction x 93 Channels.
                              These channels are expressed as a duration in
                              seconds in object DIM3_TOF, and for more details
                              see the TOF_CHANNEL_TO_SECONDS_HLC_V04.CSV file
                              in the CALIB directory of this PDS archive.
                              The Level 2 data had 96 channels, those last 3
                              are now objects TOF_WITH_START_OVERLOAD,
                              TOF_TOO_SHORT and TOF_TOO_LONG respectively."
/* RJW, DATA, f, 3, 64, 1, 93 */
      END_OBJECT           = COLUMN
    END_OBJECT           = CONTAINER
  END_OBJECT           = CONTAINER
END_OBJECT           = CONTAINER

OBJECT               = CONTAINER
  NAME               = DATA_SIGMA_DIM1
  START_BYTE         = 24129
  BYTES              = 372 /* = 1 * 93 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "DATA_SIGMA_DIM1,
                        3D array of data, 1st, 2nd and 3rd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DATA_SIGMA_DIM2
    START_BYTE         = 1
    BYTES              = 372 /* = 93 * 4-bytes */
    REPETITIONS        = 1
    DESCRIPTION        = "DATA_SIGMA_DIM2,
                          2D array of data, 2nd Dimension."
```

```
    OBJECT              = CONTAINER
      NAME              = DATA_SIGMA_DIM3
      START_BYTE        = 1
      BYTES             = 4
      REPETITIONS       = 93
      DESCRIPTION       = "DATA_SIGMA_DIM3,
                         1D array of data, 3rd Dimension."

      OBJECT              = COLUMN
        NAME              = DATA_SIGMA
        DATA_TYPE         = PC_REAL
        START_BYTE        = 1
        ITEMS             = 1
        ITEM_BYTES        = 4
        BYTES             = 4
        VALID_MINIMUM     =       0
        VALID_MAXIMUM     =  100000
        MISSING_CONSTANT  = -999999
        UNIT              = "COUNTS/SECOND"
        DESCRIPTION       = "DATA_SIGMA
                             1-sigma uncertainties on values in object DATA,
                             such that true value = DATA +/- DATA_SIGMA.
                             See DATA entry above for size information."
/* RJW, DATA_SIGMA, f, 3, 64, 1, 93 */
      END_OBJECT          = COLUMN
    END_OBJECT          = CONTAINER
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = BACKGROUND_DIM1
  START_BYTE        = 47937
  BYTES             = 372 /* = 1 * 93 * 4-bytes */
  REPETITIONS       = 64
  DESCRIPTION       = "BACKGROUND_DIM1,
                       3D array of data, 1st, 2nd and 3rd Dimensions."

  OBJECT              = CONTAINER
    NAME              = BACKGROUND_DIM2
    START_BYTE        = 1
    BYTES             = 372 /* = 93 * 4-bytes */
    REPETITIONS       = 1
    DESCRIPTION       = "BACKGROUND_DIM2,
                         2D array of data, 2nd Dimension."

    OBJECT              = CONTAINER
      NAME              = BACKGROUND_DIM3
      START_BYTE        = 1
      BYTES             = 4
      REPETITIONS       = 93
      DESCRIPTION       = "BACKGROUND_DIM3,
                         1D array of data, 3rd Dimension."

      OBJECT              = COLUMN
        NAME              = BACKGROUND
        DATA_TYPE         = PC_REAL
        START_BYTE        = 1
        ITEMS             = 1
        ITEM_BYTES        = 4
        BYTES             = 4
        VALID_MINIMUM     =       0
        VALID_MAXIMUM     = 2250000
```

```
        MISSING_CONSTANT  = -999999
        UNIT              = "COUNTS/SECOND"
        DESCRIPTION       = "Background value removed from DATA.
                             No further background removal is required.
                             If you wish to do your own background removal,
                             add this object to DATA then you can remove a
                             background via your own method.
                             The background values here were found from either
                             a background anode or JADE's own ground method."
/* RJW, BACKGROUND, f, 3, 64, 1, 93 */
      END_OBJECT          = COLUMN
    END_OBJECT          = CONTAINER
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = BACKGROUND_SIGMA_DIM1
  START_BYTE        = 71745
  BYTES             = 372 /* = 1 * 93 * 4-bytes */
  REPETITIONS       = 64
  DESCRIPTION       = "BACKGROUND_SIGMA_DIM1,
                       3D array of data, 1st, 2nd and 3rd Dimensions."

  OBJECT              = CONTAINER
    NAME              = BACKGROUND_SIGMA_DIM2
    START_BYTE        = 1
    BYTES             = 372 /* = 93 * 4-bytes */
    REPETITIONS       = 1
    DESCRIPTION       = "BACKGROUND_SIGMA_DIM2,
                         2D array of data, 2nd Dimension."

    OBJECT              = CONTAINER
      NAME              = BACKGROUND_SIGMA_DIM3
      START_BYTE        = 1
      BYTES             = 4
      REPETITIONS       = 93
      DESCRIPTION       = "BACKGROUND_SIGMA_DIM3,
                         1D array of data, 3rd Dimension."

      OBJECT              = COLUMN
        NAME              = BACKGROUND_SIGMA
        DATA_TYPE         = PC_REAL
        START_BYTE        = 1
        ITEMS             = 1
        ITEM_BYTES        = 4
        BYTES             = 4
        VALID_MINIMUM     =       0
        VALID_MAXIMUM     =  100000
        MISSING_CONSTANT  = -999999
        UNIT              = "COUNTS/SECOND"
        DESCRIPTION       = "BACKGROUND_SIGMA
                             1-sigma uncertainties on values in object
                             BACKGROUND, such that
                               true value = BACKGROUND +/- BACKGROUND_SIGMA.
                             See BACKGROUND entry above for size information."
/* RJW, BACKGROUND_SIGMA, f, 3, 64, 1, 93 */
      END_OBJECT          = COLUMN
    END_OBJECT          = CONTAINER
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = DIM1_E_DIM1
```

```
  START_BYTE        = 95553
  BYTES             = 4
  REPETITIONS       = 64
  DESCRIPTION       = "DIM1_E_DIM1,
                       (2D array of size 64x1 = 1D array of size 64.)"

      OBJECT              = COLUMN
        NAME              = DIM1_E
        DATA_TYPE         = PC_REAL
        START_BYTE        = 1
        ITEMS             = 1
        ITEM_BYTES        = 4
        BYTES             = 4
        VALID_MINIMUM     =     0.0
        VALID_MAXIMUM     = 50000.0 /* Rounded up to whole keV/q */
        MISSING_CONSTANT  = 99999.0
        UNIT              = "eV/q"
        DESCRIPTION       = "1st Dimension of DATA: Energy (center) in eV/q."
/* RJW, DIM1_E, f, 1, 64 */
      END_OBJECT          = COLUMN
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = DIM2_ELEVATION_DIM1
  START_BYTE        = 95809
  BYTES             = 4
  REPETITIONS       = 64
  DESCRIPTION       = "DIM2_ELEVATION_DIM1,
                       (2D array of size 64x1 = 1D array of size 64.)"

      OBJECT              = COLUMN
        NAME              = DIM2_ELEVATION
        DATA_TYPE         = PC_REAL
        START_BYTE        = 1
        ITEMS             = 1
        ITEM_BYTES        = 4
        BYTES             = 4
        VALID_MINIMUM     =   -90.0
                          /* 12 ion anodes cover 270 degs of elevation */
        VALID_MAXIMUM     =   180.0
        MISSING_CONSTANT  = 65535.0
        UNIT              = "Degrees"
        DESCRIPTION       = "2nd Dimension of DATA: Spacecraft elevation -
                             center value. Spacecraft elevation (degs) is
                             analogous to latitude on a sphere. In spacecraft
                             xyz co-ords:
                              +z is equivalent to elevation = +90 degs
                              -z is equivalent to elevation = -90 degs
                                (The communication dish is directed along +z)
                              xy-plane at z = 0 is equivalent to elevation = 0

                             Note, 2nd dimension is really look direction
                             which has an elevation and azimuth; hence two
                             objects describe this: DIM2_ELEVATION and
                             DIM2_AZIMUTH_DESPUN."
/* RJW, DIM2_ELEVATION, f, 1, 64 */
      END_OBJECT          = COLUMN
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = DIM2_AZIMUTH_DESPUN_DIM1
  START_BYTE        = 96065
  BYTES             = 4
```

```
  REPETITIONS       = 64
  DESCRIPTION       = "DIM2_AZIMUTH_DESPUN_DIM1,
                      (2D array of size 64x1 = 1D array of size 64.)"

      OBJECT              = COLUMN
        NAME              = DIM2_AZIMUTH_DESPUN
        DATA_TYPE         = PC_REAL
        START_BYTE        = 1
        ITEMS             = 1
        ITEM_BYTES        = 4
        BYTES             = 4
        VALID_MINIMUM     =     0.0
        VALID_MAXIMUM     =   360.0
        MISSING_CONSTANT  = 65535.0
        UNIT              = "Degrees"
        DESCRIPTION       = "2nd Dimension of DATA: Despun S/C azimuth -
                            center value. Spacecraft azimuth (degs) is
                            analogous to longitude on a sphere. In spacecraft
                            xyz co-ords:
                              +x is equivalent to azimuth =   0 degs
                              +y is equivalent to azimuth =  90 degs
                              -x is equivalent to azimuth = 180 degs
                              -y is equivalent to azimuth = 270 degs
                              +x is equivalent to azimuth = 360 degs
                              +y is equivalent to azimuth = 450 degs
                            The 'Despun' azimuth angle varies because Juno
                            spins, where azimuth = 0 is defined as +x when
                            spin phase equals zero (e.g., despun x-z plane
                            contains the ECLIPJ2000 north).

                            The relationship between despun azimuth and spin
                            phase is simply:
                               Despun Azimuth = 360 degrees - Spin Phase

                            Note, 2nd dimension is really look direction
                            which has an elevation and azimuth; hence two
                            objects describe this: DIM2_ELEVATION and
                            DIM2_AZIMUTH_DESPUN."
/* RJW, DIM2_AZIMUTH_DESPUN, f, 1, 64 */
      END_OBJECT          = COLUMN
END_OBJECT          = CONTAINER

OBJECT              = COLUMN
  NAME              = DIM3_TOF
  DATA_TYPE         = PC_REAL
  START_BYTE        = 96321
  ITEMS             = 93
  ITEM_BYTES        = 4
  BYTES             = 372
  VALID_MINIMUM     = 0.000000000
  VALID_MAXIMUM     = 0.000000330 /* = 330e-9 = 330 ns */
  MISSING_CONSTANT  = 65535.0
  UNIT              = "SECONDS"
  DESCRIPTION       = "3rd Dimension of DATA: Time Of Flight (center) value.
                      (Seconds)"
/* RJW, DIM3_TOF, f, 1, 93 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = TOF_WITH_START_OVERLOAD
  DATA_TYPE         = PC_REAL
  START_BYTE        = 96693
  ITEMS             = 64
```

```
  ITEM_BYTES         = 4
  BYTES              = 256
  VALID_MINIMUM      = 0       /* same value as for DATA object */
  VALID_MAXIMUM      = 1000000 /* same value as for DATA object */
  MISSING_CONSTANT   = -1      /* same value as for DATA object */
  UNIT               = "COUNTS/SECOND"
  DESCRIPTION        = "TOF with start overload: Counts/Second
                        A signal pulse that is too strong (above a threshold)
                        in the electronics. Multiple start-overloads that
                        occur within a 330ns event window are counted each time
                        in the Logicals Start Overload, but only once here."
/* RJW, TOF_WITH_START_OVERLOAD, f, 1, 64 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = TOF_WITH_START_OVERLOAD_SIGMA
  DATA_TYPE          = PC_REAL
  START_BYTE         = 96949
  ITEMS              = 64
  ITEM_BYTES         = 4
  BYTES              = 256
  VALID_MINIMUM      = 0       /* same value as for DATA object */
  VALID_MAXIMUM      = 1000000 /* same value as for DATA object */
  MISSING_CONSTANT   = -1      /* same value as for DATA object */
  UNIT               = "COUNTS/SECOND"
  DESCRIPTION        = "TOF with start overload uncertainty: Counts/Second
                        1-sigma uncertainties on values in object
                        TOF_WITH_START_OVERLOAD such that true value =
                    TOF_WITH_START_OVERLOAD +/- TOF_WITH_START_OVERLOAD_SIGMA.
                        See TOF_WITH_START_OVERLOAD entry above for size
                        information."
/* RJW, TOF_WITH_START_OVERLOAD_SIGMA, f, 1, 64 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = TOF_TOO_SHORT
  DATA_TYPE          = PC_REAL
  START_BYTE         = 97205
  ITEMS              = 64
  ITEM_BYTES         = 4
  BYTES              = 256
  VALID_MINIMUM      = 0       /* same value as for DATA object */
  VALID_MAXIMUM      = 1000000 /* same value as for DATA object */
  MISSING_CONSTANT   = -1      /* same value as for DATA object */
  UNIT               = "COUNTS/SECOND"
  DESCRIPTION        = "TOF too short: Counts/Second
                        TOF underflow: Count of TOF measurements that did not
                        timeout, but resulted in a measurement smaller than
                        the sensor could measure."
/* RJW, TOF_TOO_SHORT, f, 1, 64 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = TOF_TOO_SHORT_SIGMA
  DATA_TYPE          = PC_REAL
  START_BYTE         = 97461
  ITEMS              = 64
  ITEM_BYTES         = 4
  BYTES              = 256
  VALID_MINIMUM      = 0       /* same value as for DATA object */
  VALID_MAXIMUM      = 1000000 /* same value as for DATA object */
  MISSING_CONSTANT   = -1      /* same value as for DATA object */
  UNIT               = "COUNTS/SECOND"
```

```
  DESCRIPTION       = "TOF too short uncertainty: Counts/Second
                        1-sigma uncertainties on values in object
                        TOF_TOO_SHORT such that true value =
                           TOF_TOO_SHORT +/- TOF_TOO_SHORT_SIGMA.
                        See TOF_TOO_SHORT entry above for size information."
/* RJW, TOF_TOO_SHORT_SIGMA, f, 1, 64 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = TOF_TOO_LONG
  DATA_TYPE         = PC_REAL
  START_BYTE        = 97717
  ITEMS             = 64
  ITEM_BYTES        = 4
  BYTES             = 256
  VALID_MINIMUM     = 0       /* same value as for DATA object */
  VALID_MAXIMUM     = 1000000 /* same value as for DATA object */
  MISSING_CONSTANT  = -1      /* same value as for DATA object */
  UNIT              = "COUNTS/SECOND"
  DESCRIPTION       = "TOF too long: Counts/Second
                        TOF overflow: Count of TOF measurements that resulted
                        in no stop signal arriving within 330ns of the start
                        signal."
/* RJW, TOF_TOO_LONG, f, 1, 64 */
END_OBJECT          = COLUMN


OBJECT              = COLUMN
  NAME              = TOF_TOO_LONG_SIGMA
  DATA_TYPE         = PC_REAL
  START_BYTE        = 97973
  ITEMS             = 64
  ITEM_BYTES        = 4
  BYTES             = 256
  VALID_MINIMUM     = 0       /* same value as for DATA object */
  VALID_MAXIMUM     = 1000000 /* same value as for DATA object */
  MISSING_CONSTANT  = -1      /* same value as for DATA object */
  UNIT              = "COUNTS/SECOND"
  DESCRIPTION       = "TOF too long uncertainty: Counts/Second
                        1-sigma uncertainties on values in object
                        TOF_TOO_LONG such that true value =
                           TOF_TOO_LONG +/- TOF_TOO_LONG_SIGMA.
                        See TOF_TOO_LONG entry above for size information."
/* RJW, TOF_TOO_LONG_SIGMA, f, 1, 64 */
END_OBJECT          = COLUMN
```

Appendix E Level 5 data record formats

This section describes the format of the Level 5 data files.

While Section 6.2.14 ("Level 5 data files") cover this to some level, the real description is within the FMT files for each product. Here are two examples in full (one binary file, one ASCII), but see the FMT files embedded in the *.LBL files.

Font below is Courier New (to equally space characters) and size 9 in order to get 78 characters to a line. This matches the PDS files that are 80 characters to a line, but the last two are \r\n.

E.1 Sample FMT file for JAD_L50_HRS_ELC_TWO_DEF_V01.FMT

```
/* Filename: Version01/JAD_L50_HRS_ELC_TWO_DEF_V01.FMT                        */
/* File written: 2022/08/09 16:32:14                                          */
/* Will code useful Python based letters to describe each object              */
/* see http://docs.python.org/library/struct.html for codes                   */
/* formats will comma separated beginning with "RJW," as key then             */
/* {NAME}, {FORMAT}, {Number of dims}, {Size Dim 1}, {Size Dim 2}, ...        */
/* where {FORMAT} is the Python code for the type, i.e. I for uint32          */
/* and there are as many Size Dim's as number of dimensions.                  */
/* Remember to remove the comment markers at either end                       */

/* RJW, BYTES_PER_RECORD, 65898 */
/* RJW, OBJECTS_PER_RECORD, 58 */

OBJECT              = COLUMN
  NAME              = DIM0_UTC
  DATA_TYPE         = DATE  /* ASCII character string */
  START_BYTE        = 1
  BYTES             = 21
  VALID_MINIMUM     = 2011-217T00:00:00.001
                      /* SC Clock 365774402:0, JUNO Launch */
  VALID_MAXIMUM     = 2026-001T00:00:00.000 /* Expect mission end in 2025 */
  MISSING_CONSTANT  = 0001-001T00:00:00.000
  DESCRIPTION       = "UTC timestamp at center (not start) of record.
                       Format is yyyy-dddTHH:MM:SS.sss
                         where yyyy = year, ddd = day of year,
                         HH = hour, MM = minute,
                         SS.sss = decimal seconds to millisecond resolution.
                       Note: Duration of record can be found in S.I. seconds
                       by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse
                       this with the ACCUMULATION_TIME object, which is the
                       number of spacecraft clock ticks for accumulation.
                       While 1 tick is approximately 1 second, it is not
                       identical."
/* RJW, DIM0_UTC, c, 1, 21 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = PACKETID
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 22
  BYTES             = 1
  VALID_MINIMUM     = 142  /* (0x8E) */
  VALID_MAXIMUM     = 142  /* (0x8E) */
  MISSING_CONSTANT  = 255  /* Unknown, or a mix of packets */
```

```
  DESCRIPTION       = "Packet ID (DPID), Data Product Identifier
                       High Rate Science - Electron
                       Two Electron sensors per record: E060 and E180.
                       (This is the same data as for JAD_L30_HRS_ELC_ALL
                       but with E300 data removed for a smaller file.)
                       PACKETID = 142 (0x8E)"
/* RJW, PACKETID, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = DIM0_UTC_UPPER
  DATA_TYPE         = DATE  /* ASCII character string */
  START_BYTE        = 23
  BYTES             = 21
  VALID_MINIMUM     = 2011-217T00:00:00.001
  VALID_MAXIMUM     = 2026-001T00:00:00.000
  MISSING_CONSTANT  = 0001-001T00:00:00.000
  DESCRIPTION       = "0th Dimension of DATA: Time - upper limit.
                         See DIM0_UTC for description."
/* RJW, DIM0_UTC_UPPER, c, 1, 21 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = PACKET_MODE
  DATA_TYPE         = LSB_INTEGER
  START_BYTE        = 44
  BYTES             = 1
  VALID_MINIMUM     = 2
  VALID_MAXIMUM     = 2
  MISSING_CONSTANT  = 127
  DESCRIPTION       = "Packet Mode, describes type of data telemetry.
                          -2 = HSK / Housekeeping Engineering (Level 2 only)
                          -1 = HVE / High Voltage Engineering (Level 2 only)
                           0 = CAL / MCP Calibration Science  (Level 2 only)
                           1 = LRS / Low Rate Science
                           2 = HRS / High Rate Science
                           3 = DRS / DeRived Science from LRS and/or HRS
                         127 = Unknown
                         254 = Wrong - but HSK, see below.    (Level 2 only)
                         255 = Wrong - but HVE, see below.    (Level 2 only)
                       (Note, this could also be calculated via PACKETID.)
                         If you have 254 or 255 then your code is incorrect,
                       check you read a signed byte, rather than unsigned."
/* RJW, PACKET_MODE, b, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = DIM0_UTC_LOWER
  DATA_TYPE         = DATE  /* ASCII character string */
  START_BYTE        = 45
  BYTES             = 21
  VALID_MINIMUM     = 2011-217T00:00:00.001
  VALID_MAXIMUM     = 2026-001T00:00:00.000
  MISSING_CONSTANT  = 0001-001T00:00:00.000
  DESCRIPTION       = "0th Dimension of DATA: Time - lower limit.
                         See DIM0_UTC for description."
/* RJW, DIM0_UTC_LOWER, c, 1, 21 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = PACKET_SPECIES
  DATA_TYPE         = LSB_INTEGER
  START_BYTE        = 66
```

```
  BYTES              = 1
  VALID_MINIMUM      = -1
  VALID_MAXIMUM      = -1
  MISSING_CONSTANT   = 127
  DESCRIPTION        = "Packet Species, describes type of plasma data.
                           -1 = electrons
                            0 = ion species 0, SP0
                            1 = ion species 1, SP1
                            2 = ion species 2, SP2
                            3 = ion species 3, SP3
                            4 = ion species 4, SP4
                            5 = ion species 5, SP5
                            6 = ion species 6, SP6
                            7 = ion species 7, SP7
                            8 = Sum of SP3, SP4 and SP5
                            9 = All ions /* or any ion, e.g., TOF and LOG */
                           10 = Single ion species derived from TOF data
                          127 = Unknown
                          255 = Wrong - but electrons, see below.
                          If you have 255 then your code is incorrect,
                        check you read a signed byte, rather than unsigned."
/* RJW, PACKET_SPECIES, b, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = ACCUMULATION_TIME
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 67
  BYTES              = 2
  VALID_MINIMUM      = 1
  VALID_MAXIMUM      = 1
  MISSING_CONSTANT   = 65535
  UNIT               = "SECONDS" /* Not S.I. Seconds, but SCLK ticks */
  DESCRIPTION        = "Accumulation Time.
                        Number of seconds over which the data in this product
                        was collected (Science Program).
                        Note: Duration of record can be found in S.I. seconds
                        by DIM0_UTC_UPPER - DIM0_UTC_LOWER. Do not confuse
                        this with the ACCUMULATION_TIME object, which is the
                        number of spacecraft clock ticks for accumulation.
                        While 1 tick is approximately 1 second, it is not
                        identical.
                        ACCUMULATION_TIME is left in spacecraft clock ticks to
                        both aid matching with the level 2 data and to help
                        filtering for data taken in a particular mode."
/* RJW, ACCUMULATION_TIME, H, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = DATA_UNITS
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 69
  BYTES              = 1
  VALID_MINIMUM      = 3
  VALID_MAXIMUM      = 3
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Data units correspond to:
                            0 = All counts in the accumulation period
                            1 = All counts divided by number of views
                            2 = Counts per second
                                /* S.I. science units: */
                            3 = Differential Energy Flux [1/( m^2 sr s     )]
                            4 = Differential Number Flux [1/( m^2 sr s    J)]
```

```
                          5 = Phase Space Density      [    m^-6   s^3    ]
                              /* Convenient (non-S.I.) science units: */
                          6 = Differential Energy Flux [1/(cm^2 sr s     )]
                          7 = Differential Number Flux [1/(cm^2 sr s  keV)]
                          8 = Phase Space Density      [   cm^-6   s^3    ]
                 /* As new products are developed this list will increase */
                 /* If a number is not listed,               */
                 /*   try a LBL/FMT file from a recent date. */
                        255 = Unknown."
/* RJW, DATA_UNITS, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SOURCE_BACKGROUND
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 70
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 4
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Source of Background values (see BACKGROUND object)
                       that have been removed from the DATA object.
                           0 = None: No background has been removed
                           1 = Background anode (electron sensors only)
                           2 = Background anode (JADE-I only)
                           3 = Derived from Background anode : Method 1:
                                Background coefficients are time independent.
                               See file in CALIB directory for description.
                           4 = Derived from Background anode : Method 2:
                                Background coefficients are per orbit.
                               See file in CALIB directory for description.
 /* As new background removal methods are developed this list will increase */
                         255 = Unknown."
/* RJW, SOURCE_BACKGROUND, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SPARE_ZEROS
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 71
  BYTES             = 1
  VALID_MINIMUM     = 0
  VALID_MAXIMUM     = 0
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Spare Zeroes.  Always zero.
                        PDS3 format required a padding byte, e.g., a 4-byte
                        integer/float will always start on the 1st or 5th
                        or 9th or 13th... byte of the record."
/* RJW, SPARE_ZEROS, B, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SOURCE_MAG
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 72
  BYTES             = 1
  VALID_MINIMUM     =   0
  VALID_MAXIMUM     =  39
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "Source of MAG data
                       Except case 0 and 1, PAYLOAD (pl) co-ordinate MAG files
                       were used at 1s (or 2s if no 1s) resolution.
                           0 = None: No MAG data in this product.
```

```
                            1 = From Juno JADE's Level 2 files.
                                (From spacecraft and therefore uncalibrated.)
                                This is independent to JADE Level 2 version
                                number as it does not change with versions.
                                [Note MAG data in JADE files may be affected
                                by the Juno time stutter.]
                           3n = Juno's MAG's Level 3 version n calibrated
                                files, e.g., 34 means version 4, so:
                             30 = From Juno MAG's Level 3 version 00 quicklook
                                  payload files.
                                  (These are temporary files not in PDS.)
                             31 = From Juno MAG's Level 3 version 01 calibrated
                                  payload files.
                             32 = From Juno MAG's Level 3 version 02 calibrated
                                  payload files.
                             Likewise, 33 to 39 being Level 3 version 3 to 9.
                          255 = Unknown.

                           If you see a number not listed above, there may be
                         later versions of MAG data - find the latest
                         available LBL file for this product and see what that
                         has listed."
/* RJW, SOURCE_MAG, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SOURCE_JADE_METAKERNEL
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 73
  BYTES              = 2
  VALID_MINIMUM      = -32767
  VALID_MAXIMUM      =  32767
  MISSING_CONSTANT   = -32768
  DESCRIPTION        = "The JADE SPICE metakernel used to get the time,
                        position, velocity, orientation and transformation
                        objects in this file.  The metakernel lists the
                        many individual spice kernels used, which are
                        archived by NAIF and not in this PDS volume.
                        The JADE SPICE metakernel may be found in the CALIB
                        directory of this PDS volume, with filenames of:
                              JAD_L30_SPICE_METAKERNEL_nnnnn.TXT
                        where nnnnn is the SOURCE_JADE_METAKERNEL object
                        number (with leading zeros and positive).
                        If any of the kernels within the metakernel are not
                        reconstructed (but reference or predicted) for the
                        time in question, this value will be negative.
                        Within the JADE PDS archive this value should always
                        be positive."
/* RJW, SOURCE_JADE_METAKERNEL, h, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SOURCE_JADE_CALIB
  DATA_TYPE          = LSB_INTEGER
  START_BYTE         = 75
  BYTES              = 2
  VALID_MINIMUM      =      0
  VALID_MAXIMUM      =  32767
  MISSING_CONSTANT   = -32768
  DESCRIPTION        = "The JADE calibration files list used to convert the
                        engineering units of Level 2 data to the scientific
                        units in this file.  Similar to the SPICE metakernel
                        list, this lists the many individual calibration files
```

```
                       used, each of which may be found in the CALIB
                       directory on this PDS volume.

                       This list may be found in the CALIB directory of this
                       PDS volume, with filenames of:
                             JAD_L30_CALIB_LIST_nnnnn.TXT
                       where nnnnn is the SOURCE_JADE_CALIB object
                       number (with leading zeros and positive).
                       If any of the calibration files listed are not final
                       at the time in question, this value will be negative.
                       (Newer calibration files will have a higher version and
                       simply be listed in a newer SOURCE_JADE_CALIB file.)
                       Within the JADE PDS archive this value should always
                       be positive. However, a version 00 file (for team use
                       or uploaded to JSOC, not PDS) may have negative values
                       with predicted positions/orientations/transformations."
/* RJW, SOURCE_JADE_CALIB, h, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = FSW_VERSION
  DATA_TYPE         = PC_REAL
  START_BYTE        = 77
  BYTES             = 4
  VALID_MINIMUM     =   0.00
  VALID_MAXIMUM     =   9.99
  MISSING_CONSTANT  = -99.99
  DESCRIPTION       = "Flight Software version used.
                       Number should be to 2 decimal places, with rounding.
                       e.g., 4.00, 4.10, 4.20. i.e., 4.1999998 means 4.20."
/* RJW, FSW_VERSION, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = LUT_VERSION
  DATA_TYPE         = PC_REAL
  START_BYTE        = 81
  BYTES             = 4
  VALID_MINIMUM     =   0.00
  VALID_MAXIMUM     =   9.99
  MISSING_CONSTANT  = -99.99
  DESCRIPTION       = "LUT (Look Up Table) Version used on JADE.
                       Number should be to 2 decimal places, with rounding.
                       e.g., 4.00, 4.10, 4.20. i.e., 4.1999998 means 4.20."
/* RJW, LUT_VERSION, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = LUT_VERSION_SUB_LETTER
  DATA_TYPE         = CHARACTER
  START_BYTE        = 85
  BYTES             = 2
  FORMAT            = "A2"
  DESCRIPTION       = "The letter (if any) associated with the energy table
                       used at the time of this record
                           -- -> No sub letter for this LUT Version
                           -A -> Sub letter is A for this LUT Version
                           -B -> Sub letter is B for this LUT Version
                           -C -> Sub letter is C for this LUT Version
                           etc.
                       For instance, the energy table files are in the CALIB
                       directory of this PDS volume, with names like:
                       LUT_4_00_ENERGY_V01.CSV
```

```
                          (LUT_VERSION 4.00, no sub letter)
                       or
                       LUT_5_01_K_ENERGY_V01.CSV
                           (LUT_VERSION 5.01, sub letter K)."
/* RJW, LUT_VERSION_SUB_LETTER, c, 1, 2 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = LUT_SWEEP_TABLE
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 87
  BYTES              = 1
  VALID_MINIMUM      =   1
  VALID_MAXIMUM      =  23
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "The sweep tables the ion sensor used.
                       A level 2 packet will report this as 0-3,
                       However, it requires 2 packets (a ping and a pong)
                       to make a level 3 record: either 0 and 1, or 2 and 3.
                       Therefore, a value of 1 (= 01) means sweep tables
                       0 and 1 were used, while a value of 23 means sweep
                       tables 2 and 3 were used.
                       This object can only have the value of 1 or 23."
/* RJW, LUT_SWEEP_TABLE, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = FILE_VERSION
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 88
  BYTES              = 1
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 1
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "The version number of the file this record came from.
                       e.g., if you loaded file
                             JAD_L30_LRS_ION_ANY_CNT_2016240_V04.DAT
                       then FILE_VERSION = 4.
                       [FILE_VERSION = 0 is never in the PDS, but is used by
                       the JADE team prior to having required calibrations.]"
/* RJW, FILE_VERSION, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_R
  DATA_TYPE          = PC_REAL
  START_BYTE         = 89
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   130.000 /* Excluding Cruise to Jupiter */
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Jupiter Radii"
  DESCRIPTION        = "Juno radial distance at time DIM0_UTC, from
                       Jupiter, in units of Jupiter Radii (Rj).
                       (1 Rj = 71492.0 km)
                       [Values may be greater than VALID_MAXIMUM
                       during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_R, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_R_UPPER
  DATA_TYPE          = PC_REAL
```

```
  START_BYTE        = 93
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   130.000 /* Excluding Cruise to Jupiter */
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Jupiter Radii"
  DESCRIPTION       = "Juno radial distance at time DIM0_UTC_UPPER, from
                       Jupiter, in units of Jupiter Radii (Rj).
                       (1 Rj = 71492.0 km)
                       SC_POS_R_UPPER could be smaller or larger than
                       SC_POS_R, depending if moving inbound or outbound.
                       [Values may be greater than VALID_MAXIMUM
                       during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_R_UPPER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_R_LOWER
  DATA_TYPE         = PC_REAL
  START_BYTE        = 97
  BYTES             = 4
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   130.000 /* Excluding Cruise to Jupiter */
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Jupiter Radii"
  DESCRIPTION       = "Juno radial distance at time DIM0_UTC_LOWER, from
                       Jupiter, in units of Jupiter Radii (Rj).
                       (1 Rj = 71492.0 km)
                       SC_POS_R_LOWER could be smaller or larger than
                       SC_POS_R, depending if moving inbound or outbound.
                       [Values may be greater than VALID_MAXIMUM
                       during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_R_LOWER, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_LAT
  DATA_TYPE         = PC_REAL
  START_BYTE        = 101
  BYTES             = 4
  VALID_MINIMUM     =   -90.000
  VALID_MAXIMUM     =    90.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno Latitude at time DIM0_UTC, in both the
                       IAU_JUPITER and JUNO_JSS frames, in units of degrees.
                       (0 = Equatorial)
                       (JUNO_JSS is a despun version of IAU_JUPITER, hence
                       they have identical latitudes.)"
/* RJW, SC_POS_LAT, f, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_POS_LAT_UPPER
  DATA_TYPE         = PC_REAL
  START_BYTE        = 105
  BYTES             = 4
  VALID_MINIMUM     =   -90.000
  VALID_MAXIMUM     =    90.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno Latitude at time DIM0_UTC_UPPER, in both the
                       IAU_JUPITER and JUNO_JSS frames, in units of degrees.
```

```
                       (0 = Equatorial)
                       SC_POS_LAT_UPPER could be smaller or larger than
                       SC_POS_LAT.
                       (JUNO_JSS is a despun version of IAU_JUPITER, hence
                       they have identical latitudes.)"
/* RJW, SC_POS_LAT_UPPER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_LAT_LOWER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 109
  BYTES              = 4
  VALID_MINIMUM      =   -90.000
  VALID_MAXIMUM      =    90.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno Latitude at time DIM0_UTC_LOWER, in both the
                       IAU_JUPITER and JUNO_JSS frames, in units of degrees.
                       (0 = Equatorial)
                       SC_POS_LAT_LOWER could be smaller or larger than
                       SC_POS_LAT.
                       (JUNO_JSS is a despun version of IAU_JUPITER, hence
                       they have identical latitudes.)"
/* RJW, SC_POS_LAT_LOWER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_LOCAL_TIME
  DATA_TYPE          = PC_REAL
  START_BYTE         = 113
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =    24.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Hours"
  DESCRIPTION        = "Juno's (jovian) Local Time at time DIM0_UTC,
                       in units of hours.
                         00 = Midnight
                         06 = Dawn
                         12 = Noon
                         18 = Dusk"
/* RJW, SC_POS_LOCAL_TIME, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_LOCAL_TIME_UPPER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 117
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =    24.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Hours"
  DESCRIPTION        = "Juno's (jovian) Local Time at time DIM0_UTC_UPPER,
                       in units of hours.
                         00 = Midnight
                         06 = Dawn
                         12 = Noon
                         18 = Dusk"
/* RJW, SC_POS_LOCAL_TIME_UPPER, f, 1, 1 */
END_OBJECT           = COLUMN
```

```
OBJECT               = COLUMN
  NAME               = SC_POS_LOCAL_TIME_LOWER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 121
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =    24.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Hours"
  DESCRIPTION        = "Juno's (jovian) Local Time at time DIM0_UTC_LOWER,
                        in units of hours.
                          00 = Midnight
                          06 = Dawn
                          12 = Noon
                          18 = Dusk"
/* RJW, SC_POS_LOCAL_TIME_LOWER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_SYSIII_ELONG
  DATA_TYPE          = PC_REAL
  START_BYTE         = 125
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's (jovian) SYSIII (East) Longitude at time
                        DIM0_UTC, in units of degrees."
/* RJW, SC_POS_SYSIII_ELONG, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_SYSIII_ELONG_UPPER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 129
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's (jovian) SYSIII (East) Longitude at time
                        DIM0_UTC_UPPER, in units of degrees."
/* RJW, SC_POS_SYSIII_ELONG_UPPER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_SYSIII_ELONG_LOWER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 133
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's (jovian) SYSIII (East) Longitude at time
                        DIM0_UTC_LOWER, in units of degrees."
/* RJW, SC_POS_SYSIII_ELONG_LOWER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_POS_JUPITER_J2000XYZ
  DATA_TYPE          = PC_REAL
```

```
  START_BYTE        = 137
  ITEMS             = 3
  ITEM_BYTES        = 4
  BYTES             = 12
  VALID_MINIMUM     = -10008880.0 /* ~ -140 Rj */
  VALID_MAXIMUM     =  10008880.0 /* ~ +140 Rj */
  MISSING_CONSTANT  =     65535.0 /* ~ +0.917 Rj */
  UNIT              = "km"
  DESCRIPTION       = "Juno position from Jupiter in J2000 Cartesian
                       co-ordinates [x,y,z] (units km).
                       [Values may be outside of VALID_MIN/MAX range (~140Rj)
                       during cruise to Jupiter before primary mission.]"
/* RJW, SC_POS_JUPITER_J2000XYZ, f, 1, 3 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_VEL_JUPITER_J2000XYZ
  DATA_TYPE         = PC_REAL
  START_BYTE        = 149
  ITEMS             = 3
  ITEM_BYTES        = 4
  BYTES             = 12
  VALID_MINIMUM     =   -70.0
  VALID_MAXIMUM     =    70.0
  MISSING_CONSTANT  = 65535.0
  UNIT              = "km/s"
  DESCRIPTION       = "Juno Velocity with respect to Jupiter in J2000
                       Cartesian co-ordinates [Vx,Vy,Vz] (units km/s)."
/* RJW, SC_VEL_JUPITER_J2000XYZ, f, 1, 3 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_VEL_ANGULAR_J2000XYZ
  DATA_TYPE         = PC_REAL
  START_BYTE        = 161
  ITEMS             = 3
  ITEM_BYTES        = 4
  BYTES             = 12
  VALID_MINIMUM     =    -1.0 /* General limit */
  VALID_MAXIMUM     =     1.0 /* General limit */
  MISSING_CONSTANT  = 65535.0
  UNIT              = "radians/s"
  DESCRIPTION       = "Juno Angular Velocity in Cartesian co-ordinates
                       [AVx,AVy,AVz] (units radians/s).
                         (This is calculated with the SPICE ckgpav command
                         where ref=J2000.  SPICE defines it as 'This is the
                         axis about which the reference frame tied to the
                         instrument is rotating in the right-handed sense.')"
/* RJW, SC_VEL_ANGULAR_J2000XYZ, f, 1, 3 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = SC_SPIN_PERIOD
  DATA_TYPE         = PC_REAL
  START_BYTE        = 173
  BYTES             = 4
  VALID_MINIMUM     =     0.0
  VALID_MAXIMUM     =    70.0
  MISSING_CONSTANT  = 65535.0
  UNIT              = "SECONDS"
  DESCRIPTION       = "Juno spin period (seconds).
                       This is not useful during spacecraft maneuvers."
/* RJW, SC_SPIN_PERIOD, f, 1, 1 */
```

```
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_SPIN_PHASE
  DATA_TYPE          = PC_REAL
  START_BYTE         = 177
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's spin phase at time DIM0_UTC,
                        in units of degrees."
/* RJW, SC_SPIN_PHASE, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_SPIN_PHASE_UPPER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 181
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's spin phase at time DIM0_UTC_UPPER,
                        in units of degrees."
/* RJW, SC_SPIN_PHASE_UPPER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SC_SPIN_PHASE_LOWER
  DATA_TYPE          = PC_REAL
  START_BYTE         = 185
  BYTES              = 4
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   360.000
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Degrees"
  DESCRIPTION        = "Juno's spin phase at time DIM0_UTC_LOWER,
                        in units of degrees."
/* RJW, SC_SPIN_PHASE_LOWER, f, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = CONTAINER
  NAME               = DESPUN_SC_TO_J2000_DIM1
  START_BYTE         = 189
  BYTES              = 12 /* = 3 * 4-bytes */
  REPETITIONS        = 3
  DESCRIPTION        = "DESPUN_SC_TO_J2000_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DESPUN_SC_TO_J2000_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 3
    DESCRIPTION        = "DESPUN_SC_TO_J2000_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = DESPUN_SC_TO_J2000
      DATA_TYPE          = PC_REAL
```

```
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =    -1.0
      VALID_MAXIMUM     =     1.0
      MISSING_CONSTANT  = 65535.0
      DESCRIPTION       = "Rotation matrix from despun spacecraft
                           co-ordinates to J2000.
                           This is a 3x3 matrix, but if read in as a 1x9
                           stream then the 1D stream is [a,b,c, d,e,f, g,h,i]
                           and the 2D matrix would be [a,b,c
                                                        d,e,f
                                                        g,h,i]"
/* RJW, DESPUN_SC_TO_J2000, f, 2, 3, 3 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = J2000_TO_JSSXYZ_DIM1
  START_BYTE        = 225
  BYTES             = 12 /* = 3 * 4-bytes */
  REPETITIONS       = 3
  DESCRIPTION       = "J2000_TO_JSSXYZ_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = J2000_TO_JSSXYZ_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 3
    DESCRIPTION       = "J2000_TO_JSSXYZ_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = J2000_TO_JSSXYZ
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =    -1.0
      VALID_MAXIMUM     =     1.0
      MISSING_CONSTANT  = 65535.0
      DESCRIPTION       = "Rotation matrix from J2000 co-ordinates to JSS xyz
                           (JSS = Jupiter-De-Spun-Sun, see SIS for details).
                           This is a 3x3 matrix, but if read in as a 1x9
                           stream then the 1D stream is [a,b,c, d,e,f, g,h,i]
                           and the 2D matrix would be [a,b,c
                                                        d,e,f
                                                        g,h,i]"
/* RJW, J2000_TO_JSSXYZ, f, 2, 3, 3 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = J2000_TO_JSSRTP_DIM1
  START_BYTE        = 261
  BYTES             = 12 /* = 3 * 4-bytes */
  REPETITIONS       = 3
  DESCRIPTION       = "J2000_TO_JSSRTP_DIM1,
```

```
                    2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = J2000_TO_JSSRTP_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 3
    DESCRIPTION       = "J2000_TO_JSSRTP_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = J2000_TO_JSSRTP
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =    -1.0
      VALID_MAXIMUM     =     1.0
      MISSING_CONSTANT  = 65535.0
      DESCRIPTION       = "Rotation matrix from J2000 co-ordinates to
                           JSS RTP, where RTP is Jupiter centered right
                           handed R-Theta-Phi.
                           (JSS = Jupiter-De-Spun-Sun, see SIS for details.)
                           This is a 3x3 matrix, but if read in as a 1x9
                           stream then the 1D stream is [a,b,c, d,e,f, g,h,i]
                           and the 2D matrix would be [a,b,c
                                                        d,e,f
                                                        g,h,i]"
/* RJW, J2000_TO_JSSRTP, f, 2, 3, 3 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = COLUMN
  NAME              = MCP_VOLTAGE
  DATA_TYPE         = PC_REAL
  START_BYTE        = 297
  ITEMS             = 2
  ITEM_BYTES        = 4
  BYTES             = 8
  VALID_MINIMUM     = -4000.000
  VALID_MAXIMUM     =  4000.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Volts"
  DESCRIPTION       = "MCP Voltages on the two electron sensors in this
                       product, E060 and E180 respectively."
/* RJW, MCP_VOLTAGE, f, 1, 2 */
END_OBJECT          = COLUMN

OBJECT            = CONTAINER
 NAME              = ISSUES_CONTAINER
 START_BYTE        = 305
 BYTES             = 4
 REPETITIONS       = 1
 DESCRIPTION       = "ISSUES_CONTAINER, size 1."
 OBJECT            = COLUMN
  NAME              = ISSUES
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 1
  ITEMS             = 1
  ITEM_BYTES        = 4
  BYTES             = 4
```

```
VALID_MINIMUM     = 0
VALID_MAXIMUM     = 4294967294
MISSING_CONSTANT  = 4294967295
DESCRIPTION       = "Issues or potential issues in this data record.
                    These are issues that can be identified within the JADE
                    packet of data itself without any external information.
                    e.g. timing issues due to the MAG time stutter, or any
                    voltage pulsing, would not be included as there are no
                    indicators to them within this JADE packet.
                      [For a more comprehensive list of potential issues
                    from internal and external sources please see the
                    Level 3 data.]

                    Level 2 issues of this JADE packet are flagged by
                    individual bits, and several may be hit.  If no issues
                    are flagged then this 4-byte unsigned integer is zero.
                    A value of 4294967295 is the MISSING_CONSTANT and means
                    that the issue status is currently unknown.

                    All bits at 0 implies all is okay as seen by this
                    packet.  If a bit is set to 1 then that bit is flagged,
                    otherwise it is set to zero and unflagged.

                    The bits are set as followed, grouped in to seriousness:

                     Not very serious issues for doing science:
                     Bit  0 = UTC time is predicted, yet to be finalized.
                     Bit  1 = Position/Orientation values predicted, yet to
                              be finalized.  Level 3 (and above) data only.
                     Bit  2 = TABLES_VERSION object was altered on the
                              ground to accurately reflect a 'commanded
                              parameter update' outside the initial
                              per-orbit commands JADE is returning.
                               [If changed, the original downlinked
                               TABLES_VERSION value can be found by cross-
                               referencing the PARAM_TABLE_VER object in the
                               JAD_L20_HSK_ALL_SHK files.  Note here the
                               PARAM_TABLE_VER value is given as a unsigned
                               integer of Hex Major-Middle-Minor, such that
                               a value of 770 decimal is in hex 0x302,
                               meaning Table Version 3.02 ]
                     Bit  3 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
                              fixed on the ground by adjusting
                              TIMESTAMP_WHOLE, TIMESTAMP_SUB, and
                              ACCUMULATION_TIME based on cross-referencing
                              JADE commanding.
                     Bit  4 = LRS/CAL Ion Species record with unobserved
                              look directions (views) populated using views
                              from neighboring record. See Bit 12 for
                              uncorrected/unpopulated description.
                              (Only possible if ACCUMULATION_TIME = 30.)
                     Bit  5 = TIMESTAMP_WHOLE/SUB adjusted on the ground
                              to mitigate any Juno time stutter effects.
                              [Other TIMESTAMPs are susceptible to the
                              onboard time stutter too, but only the JADE
                              packet TIMESTAMP_WHOLE/SUB is tracked here.]
                     Bit  6 =       Currently unused.
                     Bit  7 = Warning, a leap second occurs during the
                              accumulation period.

                     Data slightly different than expected, but can be used
                     for science with a little extra coding:
                     Bit  8 = ACCUM_TRUNCATION object flagged.
```

```
Bit  9 = Electron (HRS/LRS/CAL) MAG objects are not
         tracked, are either zeros or MISSING_CONSTANT.
           [LRS and CAL did not have MAG objects prior
            to FSW_VERSION 4.10, therefore those MAG
            objects here are set to MISSING_CONSTANT
            when FSW_VERSION < 4.10.]
Bit 10 = TIMESTAMP_WHOLE/SUB affected by a Juno
         onboard time stutter, JADE reported timestamp
         is likely 1 whole tick too large.
         [Other TIMESTAMPs are susceptible to the
         onboard time stutter too, but only the JADE
         packet TIMESTAMP_WHOLE/SUB is tracked here.]
Bit 11 =       Currently unused.
Bit 12 = LRS/CAL Ion Species record potentially has
         unobserved look directions (spin phase sectors
         or views) present in the data, meaning the
         record may not contain data for a full 4pi
         steradians field-of-view.
         Unobserved look directions have zero counts
         per view (or counts per second) in the data,
         although an observed look direction may also
         have zero counts if no ions were measured.
         Therefore there is a potential confusion over
         zero measured counts or simply unmeasured.
         e.g. if the spin period is 30.7 seconds, then
         not all of the 78 spin phase sectors will be
         sampled in 30 seconds. (Unobserved views are
         only possible if ACCUMULATION_TIME <= 30.)
         See the JADE SIS for more information.
Bit 13 = At least one anode is blanked.
         See SIS document for further information.
Bit 14 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
         warning:
           Not fixed as yet - when fixed it will
         become bit 3 of ISSUES instead.
           Level 2 data only when FSW_VERSION = 4.00,
         ACCUMULTION_TIME object is MISSING_CONSTANT.
         Also, TIMESTAMP_WHOLE:TIMESTAMP_SUB is the end
         of the packet rather than the usual start,
         see TIMESTAMP_WHOLE object for more details.
         [Only affects data from 2015-089 to 2015-115.]
Bit 15 = Electron Anodes Reversed.
           Level 2 data only when FSW_VERSION < 4.10
         and only electron packets. Electron anodes
         are reversed in order and need to be
         remapped, however electron Spin Phase data
         (LRS data) cannot be remapped. See the SIS
         document for more information about this.
         [Affects all electron data 2011 to 2015-115.]

Data very different than expected, may not be suitable
for science - use with extreme caution.
Bit 16 = Data is not from flight instrument on Juno,
           see FLIGHT_OR_STL object.
Bit 17 = MCP_NOT_AT_COMMANDED object flagged.
           Electron HRS/CAL/HVE packets use all three
         electron sensors and therefore have three
         MCP_NOT_AT_COMMANDED values per packet.
         Setting this flag means at least one of those
         three mcps is not at its commanded value.
Bit 18 = Data includes some JADE-E300 sensor data.
         (Only flagged for HRS, LRS, CAL and HVE data.)
           E300 has a high voltage power supply issue
```

```
          and reported energy steps may be incorrect.
          If E300 is off but still reported in the data
          product, it may be zeros of fill values.
Bit 19 = Ion packet abruptly truncated.
            This packet should not be used.  It had an
          ACCUMULATION_TIME = 1, ACCUM_TRUNCATION = 1
          and the DATA object is all zeros, with a
          timestamp that matches an earlier valid packet
          that was not truncated and has non-zero DATA.
          e.g. TOF and LOG example in level 2 data at
          TIMESTAMP_WHOLE of 495879710 (UTC 2015-261).
Bit 20 = MCP Dipping Triggered, in one or more sensors.
            If the sensor measures excessive counts, it
          temporarily lowers the MCP voltage to reduce
          the number of counts and protect the sensor.
          The MCP_NOT_AT_COMMANDED object is also
          flagged (Bit 17 in ISSUES) since the MCP is
          no longer at the commanded voltage.
            For HRS/CAL/HVE electrons (datasets where
          multiple sensors are on) it is possible that
          one sensor has been dipped, but the others
          are not and still providing good data.
          (First MCP dip was HRS electrons, 2017-350.)
Bit 21 = MCP Dipped sensor's DATA set to fill values.
            If MCP dipping has triggered (Bit 20 of
          ISSUES) then:  DATA and BACKGROUND objects
          (and their *_SIGMAs) have been replaced with
          MISSING_CONSTANT values.
            (Never used for Level 2 data, which has the
          counts as measured in the dipped state.)
             In addition, Bit 17 of the ISSUES object
          (i.e. MCP_NOT_AT_COMMANDED object = 1) is set
          to zero, and, if it exists, the
          MCP_NOT_AT_COMMANDED object itself is changed
          (from 1) to be 0 for the offending sensor(s).
            If the DATA object contains data from
          multiple sensors (HRS/CAL/HVE electrons) then
          only the elements of the DATA object for the
          dipped sensor are set to MISSING_CONSTANT (as
          identified by the MCP_NOT_AT_COMMANDED value
          for each sensor (prior to setting them to 0)).
            [See Bit 22 for a similar flag.]
Bit 22 = 1 or more ELC sensor DATA set to fill values.
           Affects only electron HRS/CAL/HVE products
           (i.e. products that use multiple sensors),
           and generally only when starting that mode.
          When switching to HRS/CAL/HVE from LRS, one
          JADE-E sensor is already on, and the other(s)
          have to turn on, then it takes some time for
          that sensor to reach the commanded voltage.
          For a given record, MCP_NOT_AT_COMMANDED = 0
          for one sensor but is still = 1 for others.
          That is one sensor is taking valid science
          but the other(s) are not there yet and for
          those sensors:  DATA and BACKGROUND objects
          (and their *_SIGMAs) have been replaced
          with MISSING_CONSTANT values.
            (Never used for Level 2 data, which has the
          counts as measured in the dipped state.)
             In addition, Bit 17 of the ISSUES object
          (i.e. MCP_NOT_AT_COMMANDED object = 1) is set
          to zero, and, if it exists, the
          MCP_NOT_AT_COMMANDED object itself is changed
```

```
                                  (from 1) to be 0 for the offending sensor(s).
                                    Only the elements of the DATA object for the
                                  original MCP_NOT_AT_COMMANDED = 1 sensor(s)
                                  (prior to setting them to 0) are set to
                                  MISSING_CONSTANT.
                                    [Bits 21 and 22 are essentially the same
                                  feature caused by an mcp voltage not being at
                                  the commanded value, but the reason why this
                                  is the case is different.  The treatment is
                                  identical for both Bit 21 and Bit 22.]
                         Bit 23 =       Currently unused.

                         Bit 24 =       Currently unused.
                         Bit 25 =       Currently unused.
                         Bit 26 =       Currently unused.
                         Bit 27 =       Currently unused.
                         Bit 28 =       Currently unused.
                         Bit 29 =       Currently unused.
                         Bit 30 =       Currently unused.
                         Bit 31 = Reserved for MISSING_CONSTANT use.

                         Each bit has a decimal value of 2^{bit number}, and the
                         Issues flag is the sum of 2^{flagged bit numbers}.
                         For instance, if this ISSUES flag = 131329, then in
                         binary that value is 00000000000000100000000100000001
                         showing bits 17, 8 and 0 are flagged.

                         [If a currently unused bit is set, please check the
                         latest LBL file for this product that you can find to
                         see if it now has a definition.]"

    OBJECT             = BIT_COLUMN
      NAME             = ISSUES_BITS
      BIT_DATA_TYPE    = BOOLEAN
      START_BIT        =  1
      BITS             = 32
      ITEMS            = 32
      ITEM_BITS        =  1
      MINIMUM          =  0
      MAXIMUM          =  1
      DESCRIPTION      = "See ISSUES column object for description of bits."
    END_OBJECT         = BIT_COLUMN
/* RJW, ISSUES, I, 1, 1 */
 END_OBJECT        = COLUMN
END_OBJECT        = CONTAINER

OBJECT              = COLUMN
  NAME              = TIMESTAMP_WHOLE
  DATA_TYPE         = LSB_UNSIGNED_INTEGER
  START_BYTE        = 309
  ITEMS             = 1
  ITEM_BYTES        = 4
  BYTES             = 4
  VALID_MINIMUM     = 365774402 /* 2011-Aug-05: Juno Launch */
  VALID_MAXIMUM     = 820498148 /* ~2026-Jan-01             */
  MISSING_CONSTANT  = 0
  DESCRIPTION       = "Timestamp (Whole Second) of JADE Level 2 packet
                       used to make this Level 3 record."
/* RJW, TIMESTAMP_WHOLE, I, 1, 1 */
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = TIMESTAMP_SUB
```

```
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 313
  ITEMS              = 1
  ITEM_BYTES         = 2
  BYTES              = 2
  VALID_MINIMUM      = 0
  VALID_MAXIMUM      = 65535
  MISSING_CONSTANT   = 0
  DESCRIPTION        = "Timestamp (Subsecond) of JADE Level 2 packet
                        used to make this Level 5 record.
                        A value of 0 could be valid or a MISSING_CONSTANT,
                        but should only be treated as a MISSING_CONSTANT if
                        TIMESTAMP_WHOLE is also 0."
/* RJW, TIMESTAMP_SUB, H, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = CONTAINER
  NAME               = DATA_DIM1
  START_BYTE         = 315
  BYTES              = 128 /* = 32 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "DATA_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DATA_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 32
    DESCRIPTION        = "DATA_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = DATA
      DATA_TYPE          = PC_REAL
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      /* if background removed, VALID_MINIMUM, can be <0 */
      VALID_MINIMUM      = -5.00e+14 /* = -999998/2e-09 */
      VALID_MAXIMUM      =  1.12e+15 /* = 2250000/2e-09 */
      MISSING_CONSTANT   = -999999
      UNIT               = "1/(m^2 sr s)"
      DESCRIPTION        = "DATA: Differential Energy Flux (SI units)
                            64 Energy x 32 Look Directions.
                            "
/* RJW, DATA, f, 2, 64, 32 */
    END_OBJECT           = COLUMN
  END_OBJECT           = CONTAINER
END_OBJECT           = CONTAINER

OBJECT               = CONTAINER
  NAME               = DATA_SIGMA_DIM1
  START_BYTE         = 8507
  BYTES              = 128 /* = 32 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "DATA_SIGMA_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DATA_SIGMA_DIM2
    START_BYTE         = 1
```

```
    BYTES              = 4
    REPETITIONS        = 32
    DESCRIPTION        = "DATA_SIGMA_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = DATA_SIGMA
      DATA_TYPE          = PC_REAL
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      VALID_MINIMUM      =  0
      VALID_MAXIMUM      =  5.00e+13 /* = 100000/2e-09 */
      MISSING_CONSTANT   = -999999
      UNIT               = "1/(m^2 sr s)"
      DESCRIPTION        = "DATA_SIGMA
                            1-sigma uncertainties on values in object DATA,
                            such that true value = DATA +/- DATA_SIGMA.
                            See DATA entry above for size information."
/* RJW, DATA_SIGMA, f, 2, 64, 32 */
    END_OBJECT           = COLUMN
  END_OBJECT           = CONTAINER
END_OBJECT           = CONTAINER

OBJECT               = CONTAINER
  NAME               = BACKGROUND_DIM1
  START_BYTE         = 16699
  BYTES              = 128 /* = 32 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "BACKGROUND_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = BACKGROUND_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 32
    DESCRIPTION        = "BACKGROUND_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = BACKGROUND
      DATA_TYPE          = PC_REAL
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      VALID_MINIMUM      =  0
      VALID_MAXIMUM      =  1.12e+15 /* = 2250000/2e-09 */
      MISSING_CONSTANT   = -999999
      UNIT               = "1/(m^2 sr s)"
      DESCRIPTION        = "Background value removed from DATA.
                            If you wish to do your own background removal,
                            add this object to DATA then you can remove a
                            background via your own method.
                            See the SOURCE_BACKGROUND object for the
                            background method used per record.
                            The background values here were generated from
                            a background anode or JADE's own ground method,
                            or are all zeros if no background was removed."
/* RJW, BACKGROUND, f, 2, 64, 32 */
    END_OBJECT           = COLUMN
```

```
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = BACKGROUND_SIGMA_DIM1
  START_BYTE        = 24891
  BYTES             = 128 /* = 32 * 4-bytes */
  REPETITIONS       = 64
  DESCRIPTION       = "BACKGROUND_SIGMA_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = BACKGROUND_SIGMA_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 32
    DESCRIPTION       = "BACKGROUND_SIGMA_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = BACKGROUND_SIGMA
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =  0
      VALID_MAXIMUM     =  5.00e+13 /* = 100000/2e-09 */
      MISSING_CONSTANT  = -999999
      UNIT              = "1/(m^2 sr s)"
      DESCRIPTION       = "BACKGROUND_SIGMA
                           1-sigma uncertainties on values in object
                           BACKGROUND, such that
                             true value = BACKGROUND +/- BACKGROUND_SIGMA."
/* RJW, BACKGROUND_SIGMA, f, 2, 64, 32 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = DIM1_E_DIM1
  START_BYTE        = 33083
  BYTES             = 128 /* = 32 * 4-bytes */
  REPETITIONS       = 64
  DESCRIPTION       = "DIM1_E_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = DIM1_E_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 32
    DESCRIPTION       = "DIM1_E_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = DIM1_E
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =    0.0
```

```
      VALID_MAXIMUM     = 99000.0 /* Rounded up to whole keV/q */
      MISSING_CONSTANT  = 99999.0
      UNIT              = "eV/q"
      DESCRIPTION       = "1st Dimension of DATA: Energy (center) in eV/q."
/* RJW, DIM1_E, f, 2, 64, 32 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = DIM2_ELEVATION_DIM1
  START_BYTE        = 41275
  BYTES             = 128 /* = 32 * 4-bytes */
  REPETITIONS       = 64
  DESCRIPTION       = "DIM2_ELEVATION_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
    NAME              = DIM2_ELEVATION_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 32
    DESCRIPTION       = "DIM2_ELEVATION_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = DIM2_ELEVATION
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =   -90.0
      VALID_MAXIMUM     =    90.0
      MISSING_CONSTANT  = 65535.0
      UNIT              = "Degrees"
      DESCRIPTION       = "2nd Dimension of DATA: Spacecraft elevation -
                           center value. Spacecraft elevation (degs) is
                           analogous to latitude on a sphere. In spacecraft
                           xyz co-ords:
                            +z is equivalent to elevation = +90 degs
                            -z is equivalent to elevation = -90 degs
                              (The communication dish is directed along +z)
                            xy-plane at z = 0 is equivalent to elevation = 0

                           Note, 2nd dimension is really look direction
                           which has an elevation and azimuth; hence two
                           objects describe this: DIM2_ELEVATION and
                           DIM2_AZIMUTH_DESPUN."
/* RJW, DIM2_ELEVATION, f, 2, 64, 32 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = CONTAINER
  NAME              = DIM2_AZIMUTH_DESPUN_DIM1
  START_BYTE        = 49467
  BYTES             = 128 /* = 32 * 4-bytes */
  REPETITIONS       = 64
  DESCRIPTION       = "DIM2_AZIMUTH_DESPUN_DIM1,
                       2D array of data, 1st and 2nd Dimensions."

  OBJECT              = CONTAINER
```

```
    NAME              = DIM2_AZIMUTH_DESPUN_DIM2
    START_BYTE        = 1
    BYTES             = 4
    REPETITIONS       = 32
    DESCRIPTION       = "DIM2_AZIMUTH_DESPUN_DIM2,
                         1D array of data, 2nd Dimension."

    OBJECT              = COLUMN
      NAME              = DIM2_AZIMUTH_DESPUN
      DATA_TYPE         = PC_REAL
      START_BYTE        = 1
      ITEMS             = 1
      ITEM_BYTES        = 4
      BYTES             = 4
      VALID_MINIMUM     =     0.0
      VALID_MAXIMUM     =   360.0
      MISSING_CONSTANT  = 65535.0
      UNIT              = "Degrees"
      DESCRIPTION       = "2nd Dimension of DATA: Despun S/C azimuth -
                           center value. Spacecraft azimuth (degs) is
                           analogous to longitude on a sphere. In spacecraft
                           xyz co-ords:
                             +x is equivalent to azimuth =   0 degs
                             +y is equivalent to azimuth =  90 degs
                             -x is equivalent to azimuth = 180 degs
                             -y is equivalent to azimuth = 270 degs
                             +x is equivalent to azimuth = 360 degs
                           The 'Despun' azimuth angle varies because Juno
                           spins, where azimuth = 0 is defined as +x when
                           spin phase equals zero (e.g., despun x-z plane
                           contains the ECLIPJ2000 north).

                           The relationship between despun azimuth and spin
                           phase is simply:
                              Despun Azimuth = 360 degrees - Spin Phase

                           Note, 2nd dimension is really look direction
                           which has an elevation and azimuth; hence two
                           objects describe this: DIM2_ELEVATION and
                           DIM2_AZIMUTH_DESPUN."
/* RJW, DIM2_AZIMUTH_DESPUN, f, 2, 64, 32 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER

OBJECT              = COLUMN
  NAME              = MAG_UTC
  DATA_TYPE         = DATE  /* ASCII character string */
  START_BYTE        = 57659
  BYTES             = 21
  VALID_MINIMUM     = 2011-217T00:00:00.001
                      /* SC Clock 365774402:0, JUNO Launch */
  VALID_MAXIMUM     = 2026-001T00:00:00.000 /* Expect mission end in 2025 */
  MISSING_CONSTANT  = 0001-001T00:00:00.000
  DESCRIPTION       = "UTC timestamp of MAG record to be used for MAG
                       related objects in this JADE record.
                       Format is yyyy-dddTHH:MM:SS.sss
                         where yyyy = year, ddd = day of year,
                         HH = hour, MM = minute,
                         SS.sss = decimal seconds to millisecond resolution.

                       Generally this is the 1s resolution MAG data record
                       closest to DIM0_UTC. For Low Rate Science data, this
```

```
                       must be true:
                         DIM0_UTC_LOWER <= MAG_UTC <= DIM0_UTC_UPPER
                       or else MISSING_CONSTANT values are used in all MAG
                       objects.
                       For High Rate Science data, it must be within 30s
                       (~1 spin) so that this must be true:
                         DIM0_UTC - 15s <= MAG_UTC <= DIM0_UTC + 15s
                       or else MISSING_CONSTANT values are used in all MAG
                       objects."
/* RJW, MAG_UTC, c, 1, 21 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SOURCE_JADE_LEVEL3_VERSION_INPUT
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 57680
  BYTES              = 1
  VALID_MINIMUM      =   4
  VALID_MAXIMUM      = 254
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Version number (nn) of Level 3 file JAD_L30_*_CNT_*Vnn
                       file used as input to create this Level 5 file.
                       e.g. if file JAD_L50_HRS_ELC_TWO_DEF_yyyyddd_V01
                       was generated from JAD_L30_HRS_ELC_TWO_CNT_yyyyddd_V04
                       then SOURCE_JADE_LEVEL3_VERSION_INPUT = 4 (and
                       FILE_VERSION = 1)."
/* RJW, SOURCE_JADE_LEVEL3_VERSION_INPUT, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = SOURCE_JADE_GFXEFF_VERSION
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 57681
  BYTES              = 1
  VALID_MINIMUM      =   1
  VALID_MAXIMUM      = 254
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Version number of the Geometric Factor * EFFiciency
                       calculation for this record (time dependent)."
/* RJW, SOURCE_JADE_GFXEFF_VERSION, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = MAG_RANGE
  DATA_TYPE          = LSB_UNSIGNED_INTEGER
  START_BYTE         = 57682
  BYTES              = 1
  VALID_MINIMUM      =   0
  VALID_MAXIMUM      =   6
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "MAG instrument range (0-6).
                       From the MAG PDS files, described as: [MAG] Instrument
                       dynamic range identifier at time of the sample.
                       See the Level 3 MAG SIS for further detail."
/* RJW, MAG_RANGE, B, 1, 1 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = MAG_VECTOR_DESPUN
  DATA_TYPE          = PC_REAL
  START_BYTE         = 57683
  ITEMS              = 3
  ITEM_BYTES         = 4
```

```
  BYTES              = 12
  VALID_MINIMUM      = -1600000.0
  VALID_MAXIMUM      =  1600000.0
  MISSING_CONSTANT   =  9990000.0
  UNIT               = "nT"
  DESCRIPTION        = "Despun MAG vector in nT, 3 components [X, Y, Z] at
                        time MAG_UTC.
                        MAG range is +/- 16 G (= 1600000 nT), hence limits.
                        This xyz coordinate system is despun spacecraft; see
                        the definitions of DIM2_ELEVATION and DIM2_AZIMUTH:
                          +X is when [azimuth, elevation] = [  0,   0] degrees,
                          +Y is when [azimuth, elevation] = [ 90,  0] degrees,
                          +Z is when elevation = 90 degrees."
/* RJW, MAG_VECTOR_DESPUN, f, 1, 3 */
END_OBJECT           = COLUMN

OBJECT               = COLUMN
  NAME               = MAG_VECTOR_JSSRTP
  DATA_TYPE          = PC_REAL
  START_BYTE         = 57695
  ITEMS              = 3
  ITEM_BYTES         = 4
  BYTES              = 12
  VALID_MINIMUM      = -1600000.0
  VALID_MAXIMUM      =  1600000.0
  MISSING_CONSTANT   =  9990000.0
  UNIT               = "nT"
  DESCRIPTION        = "MAG vector in JSS spherical components [Br, Bth, Bphi]
                        in nT at time MAG_UTC, and is intended as a guide for
                        context only. If you wish to do science with the MAG
                        data, please use the PDS MAG datasets, and not this
                        down-sampled vector used for the JADE data.
                        This vector is identical to System III spherical
                        components, since the JUNO_JSS frame has the
                        same spin-axis as System III (IAU_JUPITER)."
/* RJW, MAG_VECTOR_JSSRTP, f, 1, 3 */
END_OBJECT           = COLUMN

OBJECT               = CONTAINER
  NAME               = DIM3_PITCH_ANGLES_DIM1
  START_BYTE         = 57707
  BYTES              = 128 /* = 32 * 4-bytes */
  REPETITIONS        = 64
  DESCRIPTION        = "DIM3_PITCH_ANGLES_DIM1,
                        2D array of data, 1st and 2nd Dimensions."

  OBJECT               = CONTAINER
    NAME               = DIM3_PITCH_ANGLES_DIM2
    START_BYTE         = 1
    BYTES              = 4
    REPETITIONS        = 32
    DESCRIPTION        = "DIM3_PITCH_ANGLES_DIM2,
                          1D array of data, 2nd Dimension."

    OBJECT               = COLUMN
      NAME               = DIM3_PITCH_ANGLES
      DATA_TYPE          = PC_REAL
      START_BYTE         = 1
      ITEMS              = 1
      ITEM_BYTES         = 4
      BYTES              = 4
      VALID_MINIMUM      =     0
      VALID_MAXIMUM      =   180
```

```
      MISSING_CONSTANT  = 65535
      UNIT              = "Degrees"
      DESCRIPTION       = "Pitch Angles of each element of the DATA
                           object. The MAG vector provided in
                           MAG_VECTOR_DESPUN was used, and it is assumed
                           that that MAG vector is constant over the
                           whole accumulation period."
/* RJW, DIM3_PITCH_ANGLES, f, 2, 64, 32 */
    END_OBJECT          = COLUMN
  END_OBJECT          = CONTAINER
END_OBJECT          = CONTAINER
```

E.2 Sample FMT file for JAD_L50_HLS_ION_MOM_ISO_3D_PROTONS_V01.FMT

```
/* Filename: Version01/JAD_L50_HLS_ION_MOM_ISO_3D_PROTONS_V01.FMT           */
/* File written: 2022/08/09 16:32:14                                        */
/* Will code useful Python based letters to describe each object            */
/* see http://docs.python.org/library/struct.html for codes                 */
/* formats will comma separated beginning with "RJW," as key then           */
/* {NAME}, {FORMAT}, {Number of dims}, {Size Dim 1}, {Size Dim 2}, ...      */
/* where {FORMAT} is the Python code for the type, i.e. I for uint32        */
/* and there are as many Size Dim's as number of dimensions.                */
/* Remember to remove the comment markers at either end                     */
/*                                                                          */
/* The above (RJW tags) is for binary files, but for CSV files, we use the  */
/* tag RJWcsv. Now the {FORMAT} code is the what you should convert the     */
/* text value to. e.g. {FORMAT} of d means a value 5 should be read in as   */
/* a double. (And BYTES_PER_RECORD really means characters per record.)     */

/* RJWcsv, BYTES_PER_RECORD, 377 */
/* RJWcsv, OBJECTS_PER_RECORD, 27 */

OBJECT              = FIELD
  NAME              = "UTC"
  DATA_TYPE         = DATE  /* ASCII character string */
  FIELD_NUMBER      = 1
  BYTES             = 21
  VALID_MINIMUM     = 2011-217T00:00:00.001
                      /* SC Clock 365774402:0, JUNO Launch */
  VALID_MAXIMUM     = 2026-001T00:00:00.000 /* Expect mission end in 2025 */
  MISSING_CONSTANT  = 0001-001T00:00:00.000
  DESCRIPTION       = "UTC timestamp at center (not start) of record.
                     Format is yyyy-dddTHH:MM:SS.sss
                       where yyyy = year, ddd = day of year,
                       HH = hour, MM = minute,
                       SS.sss = decimal seconds to millisecond resolution.
                     Note: Duration of record can be found in seconds from
                     ACCUMULATION_TIME.  This record really covers the
                     period starting at UTC - ACCUMULATION_TIME/2 (inclusive)
                     and ending at UTC + ACCUMULATION_TIME/2
                     Technically, ACCUMULATION_TIME is in spacecraft clock
                     ticks, where 1 tick is approximately 1 second, but is
                     so close that, practically, we consider it as seconds."
/* RJWcsv, UTC, c, 1, 21 */
END_OBJECT          = FIELD

OBJECT              = FIELD
  NAME              = "SOURCE_JADE_LEVEL5_DEF_VERSION_INPUT"
  FIELD_NUMBER      = 2
  BYTES             = 3
  FORMAT            = "I3"
  DATA_TYPE         = "ASCII_INTEGER"
  MISSING_CONSTANT  = 255
  DESCRIPTION       = "The file version of the Level 5 DEF file used as
                       input to calculate these moments. e.g. if file
                       JAD_L50_LRS_ELC_ANY_DEF_2017112_V03.DAT was used
                       then SOURCE_JADE_LEVEL5_DEF_VERSION_INPUT = 03."
/* RJWcsv, SOURCE_JADE_LEVEL5_DEF_VERSION_INPUT, I, 1, 1 */
END_OBJECT          = FIELD

OBJECT              = FIELD
  NAME              = "INPUT_DATA_SELECTION"
```

```
  FIELD_NUMBER      = 3
  BYTES             = 3
  FORMAT            = "I3"
  DATA_TYPE         = "ASCII_INTEGER"
  VALID_MINIMUM     =   3
  VALID_MAXIMUM     = 345
  MISSING_CONSTANT  = 999
  DESCRIPTION       = "Input Data Selection: which subset of input data was
                       used to generate the moments?
                       This is a simple look up table:
                         3 = ion species 3 only
                         4 = ion species 4 only
                         5 = ion species 5 only
                        34 = ion species 3 and 4 combined
                        45 = ion species 4 and 5 combined
                        60 = JADE-E060 only (electrons)
                       103 = ion species 3 only with TOF correction
                       104 = ion species 4 only with TOF correction
                       105 = ion species 5 only with TOF correction
                       134 = ion species 3 and 4 combined with TOF correction
                       145 = ion species 4 and 5 combined with TOF correction
                       180 = JADE-E180 only (electrons)
                       240 = JADE-E060 and JADE-E180 combined (electrons)
                       345 = ion species 3, 4 and 5 combined
                       999 = MISSING_CONSTANT = Unknown

                       [Other entries may be added later as new techniques
                       are explored/used. If your number is not listed here,
                       try looking in the LBL file description of the latest
                       file.]

                       The TOF correction would account for false coincidence
                       counts falling in other ion species datasets."
/* RJWcsv, INPUT_DATA_SELECTION, I, 1, 1 */
END_OBJECT          = FIELD

OBJECT              = FIELD
  NAME              = "PACKET_MODE"
  FIELD_NUMBER      = 4
  BYTES             = 3
  FORMAT            = "I3"
  DATA_TYPE         = "ASCII_INTEGER"
  VALID_MINIMUM     =   1
  VALID_MAXIMUM     =   2
  MISSING_CONSTANT  = 127
  DESCRIPTION       = "Packet Mode, describes type of data telemetry.
                           1 = LRS / Low Rate Science
                           2 = HRS / High Rate Science
                         127 = Unknown"
/* RJWcsv, PACKET_MODE, I, 1, 1 */
END_OBJECT          = FIELD

OBJECT              = FIELD
  NAME              = "ACCUMULATION_TIME"
  FIELD_NUMBER      = 5
  BYTES             = 5
  FORMAT            = "I5"
  DATA_TYPE         = "ASCII_INTEGER"
  UNIT              = "SECONDS" /* Not S.I. Seconds, but SCLK ticks */
  VALID_MINIMUM     =     1
  VALID_MAXIMUM     =  1800
  MISSING_CONSTANT  = 65535
  DESCRIPTION       = "Accumulation Time.
```

```
                       Number of spacecraft clock ticks (assume seconds)
                       over which the data in this product was collected.
                       While 1 tick is approximately 1 second, it is not
                       identical, but close enough that it is assumed to be."
/* RJWcsv, ACCUMULATION_TIME, I, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "SOURCE_BACKGROUND"
  FIELD_NUMBER       = 6
  BYTES              = 3
  FORMAT             = "I3"
  DATA_TYPE          = "ASCII_INTEGER"
  VALID_MINIMUM      =   0
  VALID_MAXIMUM      =   4
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Source of Background values
                       that have been removed from the DATA object.
                           0 = None: No background has been removed
                           1 = Background anode (electron sensors only)
                           2 = Background anode (JADE-I only)
                           3 = Derived from Background anode : Method 1:
                                Background coefficients are time independent.
                               See file in CALIB directory for description.
                           4 = Derived from Background anode : Method 2:
                                Background coefficients are per orbit.
                               See file in CALIB directory for description.
 /* As new background removal methods are developed this list will increase */
                         255 = Unknown."
/* RJWcsv, SOURCE_BACKGROUND, I, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "ISSUES"
  FIELD_NUMBER       = 7
  BYTES              = 10
  FORMAT             = "I10"
  DATA_TYPE          = "ASCII_INTEGER"
  VALID_MINIMUM      =          0
  VALID_MAXIMUM      = 4294967294
  MISSING_CONSTANT   = 4294967295
  DESCRIPTION        = "Issues or potential issues in this data record.
                       [Level 3 ion records have a ping and pong half, each
                       with an ISSUES value.  These two values have been
                       merged with a bitwise OR to give a single value in
                       this file.]

                      These are issues that can be identified within the JADE
                      packet of data itself without any external information.
                      e.g. timing issues due to the MAG time stutter, or any
                      voltage pulsing, would not be included as there are no
                      indicators to them within this JADE packet.
                        [For a more comprehensive list of potential issues
                      from internal and external sources please see the
                      Level 3 data.]

                      Level 2 issues of this JADE packet are flagged by
                      individual bits, and several may be hit.  If no issues
                      are flagged then this 4-byte unsigned integer is zero.
                      A value of 4294967295 is the MISSING_CONSTANT and means
                      that the issue status is currently unknown.

                      All bits at 0 implies all is okay as seen by this
```

```
packet.  If a bit is set to 1 then that bit is flagged,
otherwise it is set to zero and unflagged.

The bits are set as followed, grouped in to seriousness:

 Not very serious issues for doing science:
 Bit  0 = UTC time is predicted, yet to be finalized.
 Bit  1 = Position/Orientation values predicted, yet to
          be finalized.  Level 3 (and above) data only.
 Bit  2 = TABLES_VERSION object was altered on the
          ground to accurately reflect a 'commanded
          parameter update' outside the initial
          per-orbit commands JADE is returning.
           [If changed, the original downlinked
           TABLES_VERSION value can be found by cross-
           referencing the PARAM_TABLE_VER object in the
           JAD_L20_HSK_ALL_SHK files.  Note here the
           PARAM_TABLE_VER value is given as a unsigned
           integer of Hex Major-Middle-Minor, such that
           a value of 770 decimal is in hex 0x302,
           meaning Table Version 3.02 ]
 Bit  3 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
          fixed on the ground by adjusting
          TIMESTAMP_WHOLE, TIMESTAMP_SUB, and
          ACCUMULATION_TIME based on cross-referencing
          JADE commanding.
 Bit  4 = LRS/CAL Ion Species record with unobserved
          look directions (views) populated using views
          from neighboring record. See Bit 12 for
          uncorrected/unpopulated description.
          (Only possible if ACCUMULATION_TIME = 30.)
 Bit  5 = TIMESTAMP_WHOLE/SUB adjusted on the ground
          to mitigate any Juno time stutter effects.
          [Other TIMESTAMPs are susceptible to the
          onboard time stutter too, but only the JADE
          packet TIMESTAMP_WHOLE/SUB is tracked here.]
 Bit  6 =       Currently unused.
 Bit  7 = Warning, a leap second occurs during the
          accumulation period.

 Data slightly different than expected, but can be used
 for science with a little extra coding:
 Bit  8 = ACCUM_TRUNCATION object flagged.
 Bit  9 = Electron (HRS/LRS/CAL) MAG objects are not
          tracked, are either zeros or MISSING_CONSTANT.
            [LRS and CAL did not have MAG objects prior
             to FSW_VERSION 4.10, therefore those MAG
             objects here are set to MISSING_CONSTANT
             when FSW_VERSION < 4.10.]
 Bit 10 = TIMESTAMP_WHOLE/SUB affected by a Juno
          onboard time stutter, JADE reported timestamp
          is likely 1 whole tick too large.
          [Other TIMESTAMPs are susceptible to the
          onboard time stutter too, but only the JADE
          packet TIMESTAMP_WHOLE/SUB is tracked here.]
 Bit 11 =       Currently unused.
 Bit 12 = LRS/CAL Ion Species record potentially has
          unobserved look directions (spin phase sectors
          or views) present in the data, meaning the
          record may not contain data for a full 4pi
          steradians field-of-view.
          Unobserved look directions have zero counts
          per view (or counts per second) in the data,
```

```
          although an observed look direction may also
          have zero counts if no ions were measured.
          Therefore there is a potential confusion over
          zero measured counts or simply unmeasured.
          e.g. if the spin period is 30.7 seconds, then
          not all of the 78 spin phase sectors will be
          sampled in 30 seconds. (Unobserved views are
          only possible if ACCUMULATION_TIME <= 30.)
          See the JADE SIS for more information.
Bit 13 = At least one anode is blanked.
          See SIS document for further information.
Bit 14 = FSW_VERSION 4.00 LRS/CAL Ion Species bug
          warning:
            Not fixed as yet - when fixed it will
          become bit 3 of ISSUES instead.
            Level 2 data only when FSW_VERSION = 4.00,
          ACCUMULTION_TIME object is MISSING_CONSTANT.
          Also, TIMESTAMP_WHOLE:TIMESTAMP_SUB is the end
          of the packet rather than the usual start,
          see TIMESTAMP_WHOLE object for more details.
          [Only affects data from 2015-089 to 2015-115.]
Bit 15 = Electron Anodes Reversed.
            Level 2 data only when FSW_VERSION < 4.10
          and only electron packets. Electron anodes
          are reversed in order and need to be
          remapped, however electron Spin Phase data
          (LRS data) cannot be remapped. See the SIS
          document for more information about this.
          [Affects all electron data 2011 to 2015-115.]

Data very different than expected, may not be suitable
for science - use with extreme caution.
Bit 16 = Data is not from flight instrument on Juno,
            see FLIGHT_OR_STL object.
Bit 17 = MCP_NOT_AT_COMMANDED object flagged.
            Electron HRS/CAL/HVE packets use all three
          electron sensors and therefore have three
          MCP_NOT_AT_COMMANDED values per packet.
          Setting this flag means at least one of those
          three mcps is not at its commanded value.
Bit 18 = Data includes some JADE-E300 sensor data.
          (Only flagged for HRS, LRS, CAL and HVE data.)
            E300 has a high voltage power supply issue
          and reported energy steps may be incorrect.
          If E300 is off but still reported in the data
          product, it may be zeros of fill values.
Bit 19 = Ion packet abruptly truncated.
            This packet should not be used.  It had an
          ACCUMULATION_TIME = 1, ACCUM_TRUNCATION = 1
          and the DATA object is all zeros, with a
          timestamp that matches an earlier valid packet
          that was not truncated and has non-zero DATA.
          e.g. TOF and LOG example in level 2 data at
          TIMESTAMP_WHOLE of 495879710 (UTC 2015-261).
Bit 20 = MCP Dipping Triggered, in one or more sensors.
            If the sensor measures excessive counts, it
          temporarily lowers the MCP voltage to reduce
          the number of counts and protect the sensor.
          The MCP_NOT_AT_COMMANDED object is also
          flagged (Bit 17 in ISSUES) since the MCP is
          no longer at the commanded voltage.
            For HRS/CAL/HVE electrons (datasets where
          multiple sensors are on) it is possible that
```

```
          one sensor has been dipped, but the others
          are not and still providing good data.
          (First MCP dip was HRS electrons, 2017-350.)
Bit 21 = MCP Dipped sensor's DATA set to fill values.
           If MCP dipping has triggered (Bit 20 of
          ISSUES) then:  DATA and BACKGROUND objects
          (and their *_SIGMAs) have been replaced with
          MISSING_CONSTANT values.
           (Never used for Level 2 data, which has the
          counts as measured in the dipped state.)
            In addition, Bit 17 of the ISSUES object
          (i.e. MCP_NOT_AT_COMMANDED object = 1) is set
          to zero, and, if it exists, the
          MCP_NOT_AT_COMMANDED object itself is changed
          (from 1) to be 0 for the offending sensor(s).
           If the DATA object contains data from
          multiple sensors (HRS/CAL/HVE electrons) then
          only the elements of the DATA object for the
          dipped sensor are set to MISSING_CONSTANT (as
          identified by the MCP_NOT_AT_COMMANDED value
          for each sensor (prior to setting them to 0)).
           [See Bit 22 for a similar flag.]
Bit 22 = 1 or more ELC sensor DATA set to fill values.
           Affects only electron HRS/CAL/HVE products
           (i.e. products that use multiple sensors),
           and generally only when starting that mode.
          When switching to HRS/CAL/HVE from LRS, one
          JADE-E sensor is already on, and the other(s)
          have to turn on, then it takes some time for
          that sensor to reach the commanded voltage.
          For a given record, MCP_NOT_AT_COMMANDED = 0
          for one sensor but is still = 1 for others.
          That is one sensor is taking valid science
          but the other(s) are not there yet and for
          those sensors:  DATA and BACKGROUND objects
          (and their *_SIGMAs) have been replaced
          with MISSING_CONSTANT values.
           (Never used for Level 2 data, which has the
          counts as measured in the dipped state.)
            In addition, Bit 17 of the ISSUES object
          (i.e. MCP_NOT_AT_COMMANDED object = 1) is set
          to zero, and, if it exists, the
          MCP_NOT_AT_COMMANDED object itself is changed
          (from 1) to be 0 for the offending sensor(s).
           Only the elements of the DATA object for the
          original MCP_NOT_AT_COMMANDED = 1 sensor(s)
          (prior to setting them to 0) are set to
          MISSING_CONSTANT.
           [Bits 21 and 22 are essentially the same
          feature caused by an mcp voltage not being at
          the commanded value, but the reason why this
          is the case is different.  The treatment is
          identical for both Bit 21 and Bit 22.]
Bit 23 =       Currently unused.

Bit 24 =       Currently unused.
Bit 25 =       Currently unused.
Bit 26 =       Currently unused.
Bit 27 =       Currently unused.
Bit 28 =       Currently unused.
Bit 29 =       Currently unused.
Bit 30 =       Currently unused.
Bit 31 = Reserved for MISSING_CONSTANT use.
```

```
                       Each bit has a decimal value of 2^{bit number}, and the
                       Issues flag is the sum of 2^{flagged bit numbers}.
                       For instance, if this ISSUES flag = 131329, then in
                       binary that value is 00000000000000100000000100000001
                       showing bits 17, 8 and 0 are flagged.

                       [If a currently unused bit is set, please check the
                       latest LBL file for this product that you can find to
                       see if it now has a definition.]"
/* RJWcsv, ISSUES, I, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "EV_PER_Q_RANGE"
  FIELD_NUMBER       = 8
  BYTES              = 22
  ITEMS              = 2
  ITEM_BYTES         = 10
  FORMAT             = "F10.3"
  DATA_TYPE          = "ASCII_REAL"
  VALID_MINIMUM      =      0.000
  VALID_MAXIMUM      = 100000.000
  MISSING_CONSTANT   = -99999.000
  UNIT               = "eV/q"
  DESCRIPTION        = "Energy Range of sensor(s) (eV/q) [lower, upper].
                       Each JADE sensor has its own energy range, and these
                       do vary over time, with occasional significant
                       changes of energy tables. This object is to give
                       context to the moments, in particular, what energy
                       range were moments calculated over.
                       If two sensors data were combined, then this would
                       reflect the merged energy table limits rather than
                       of one particular sensor."
/* RJWcsv, EV_PER_Q_RANGE, d, 1, 2 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "SC_POS_R"
  FIELD_NUMBER       = 9
  BYTES              = 9
  FORMAT             = "F9.3"
  DATA_TYPE          = "ASCII_REAL"
  VALID_MINIMUM      =     0.000
  VALID_MAXIMUM      =   130.000 /* Excluding Cruise to Jupiter */
  MISSING_CONSTANT   = 65535.000
  UNIT               = "Jupiter Radii"
  DESCRIPTION        = "Juno radial distance at time UTC, from
                       Jupiter, in units of Jupiter Radii (Rj).
                       (1 Rj = 71492.0 km)
                       [Values may be greater than VALID_MAXIMUM
                       during cruise to Jupiter before primary mission.]"
/* RJWcsv, SC_POS_R, d, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "SC_POS_LAT"
  FIELD_NUMBER       = 10
  BYTES              = 9
  FORMAT             = "F9.3"
  DATA_TYPE          = "ASCII_REAL"
  VALID_MINIMUM      =   -90.000
  VALID_MAXIMUM      =    90.000
```

```
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno Latitude at time UTC, in both the
                       IAU_JUPITER and JUNO_JSS frames, in units of degrees.
                       (0 = Equatorial)
                       (JUNO_JSS is a despun version of IAU_JUPITER, hence
                       they have identical latitudes.)"
/* RJWcsv, SC_POS_LAT, d, 1, 1 */
END_OBJECT          = FIELD

OBJECT              = FIELD
  NAME              = "SC_POS_LOCAL_TIME"
  FIELD_NUMBER      = 11
  BYTES             = 9
  FORMAT            = "F9.3"
  DATA_TYPE         = "ASCII_REAL"
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =    24.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Hours"
  DESCRIPTION       = "Juno's (jovian) Local Time at time UTC,
                       in units of hours.
                         00 = Midnight
                         06 = Dawn
                         12 = Noon
                         18 = Dusk"
/* RJWcsv, SC_POS_LOCAL_TIME, d, 1, 1 */
END_OBJECT          = FIELD

OBJECT              = FIELD
  NAME              = "SC_POS_SYSIII_ELONG"
  FIELD_NUMBER      = 12
  BYTES             = 9
  FORMAT            = "F9.3"
  DATA_TYPE         = "ASCII_REAL"
  VALID_MINIMUM     =     0.000
  VALID_MAXIMUM     =   360.000
  MISSING_CONSTANT  = 65535.000
  UNIT              = "Degrees"
  DESCRIPTION       = "Juno's (jovian) SYSIII (East) Longitude at time UTC,
                       in units of degrees."
/* RJWcsv, SC_POS_SYSIII_ELONG, d, 1, 1 */
END_OBJECT          = FIELD

OBJECT              = FIELD
  NAME              = "DIMENSIONS"
  FIELD_NUMBER      = 13
  BYTES             = 1
  FORMAT            = "I1"
  DATA_TYPE         = "ASCII_INTEGER"
  VALID_MINIMUM     = 1
  VALID_MAXIMUM     = 3
  MISSING_CONSTANT  = 0
  DESCRIPTION       = "Dimensionality of moments: are these calculated
                       in 1D (=1), 2D (=2) or 3D (=3)."
/* RJWcsv, DIMENSIONS, I, 1, 1 */
END_OBJECT          = FIELD

OBJECT              = FIELD
  NAME              = "M"
  FIELD_NUMBER      = 14
  BYTES             = 9
  FORMAT            = "E9.3"
```

```
  DATA_TYPE          = "ASCII_REAL"
  VALID_MINIMUM      =  5.486E-04 /* i.e. electron mass */
  VALID_MAXIMUM      =  6.400E+01 /* i.e. 64 (S+) */
  UNIT               = "amu"
  DESCRIPTION        = "Mass of particle used for moments calculations in
                        units of amu (atomic mass units).
                        Valid minimum is 5.486E-04 amu, which is the mass of
                        an electron (and why the E9.3 format was chosen),
                        but the electron moment code actually used more
                        precision with M_e of 0.00054857990907 amu."
/* RJWcsv, M, d, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "Q"
  FIELD_NUMBER       = 15
  BYTES              = 5
  FORMAT             = "F5.2"
  DATA_TYPE          = "ASCII_REAL"
  VALID_MINIMUM      =  -1.00
  VALID_MAXIMUM      =   4.00
  UNIT               = "e"
  DESCRIPTION        = "Charge of particle used for moments calculations in
                        units of e (elementary charge).
                        i.e. an electron has a charge of -1, and a proton +1.
                        e.g. For ions that are a mix of O+ and S++, we may use
                        M = 24 and Q = 1.5, so that M/Q = 16."
/* RJWcsv, Q, d, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "NUM_LOOK_DIRS"
  FIELD_NUMBER       = 16
  BYTES              = 3
  FORMAT             = "I3"
  DATA_TYPE          = "ASCII_INTEGER"
  VALID_MINIMUM      =   1
  VALID_MAXIMUM      = 120
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Number of Look Directions used in moments
                       calculations. i.e. Low rate science ion species has 78
                       look directions, while 1D electron moments would only
                       have 1 look direction."
/* RJWcsv, NUM_LOOK_DIRS, I, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "N_CC"
  FIELD_NUMBER       = 17
  BYTES              = 10
  FORMAT             = "E10.3"
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  UNIT               = "1/(cm^3)"
  DESCRIPTION        = "Number Density in units of 1/cm^3."
/* RJWcsv, N_CC, d, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "N_SIGMA_CC"
  FIELD_NUMBER       = 18
  BYTES              = 10
  FORMAT             = "E10.3"
```

```
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  UNIT               = "1/(cm^3)"
  DESCRIPTION        = "Number Density Uncertainty in units of 1/cm^3."
/* RJWcsv, N_SIGMA_CC, d, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "V_JSSXYZ_KMPS"
  FIELD_NUMBER       = 19
  BYTES              = 34
  ITEMS              = 3
  ITEM_BYTES         = 10
  FORMAT             = "F10.3"
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  UNIT               = "km/s"
  DESCRIPTION        = "Velocity Vector in the Cartesian JUNO_JSS
                        (Jupiter-deSpun-Sun) frame in units of km/s.
                        Three components are provided:
                           [V_x, V_y, V_z]"
/* RJWcsv, V_JSSXYZ_KMPS, d, 1, 3 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "V_JSSXYZ_SIGMA_KMPS"
  FIELD_NUMBER       = 20
  BYTES              = 34
  ITEMS              = 3
  ITEM_BYTES         = 10
  FORMAT             = "F10.3"
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  UNIT               = "km/s"
  DESCRIPTION        = "Velocity Vector uncertainty in the Cartesian JUNO_JSS
                        (Jupiter-deSpun-Sun) frame in units of km/s.
                        Three components are provided:
                           [V_sigma_x, V_sigma_y, V_sigma_z]"
/* RJWcsv, V_JSSXYZ_SIGMA_KMPS, d, 1, 3 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "V_JSSRTP_KMPS"
  FIELD_NUMBER       = 21
  BYTES              = 34
  ITEMS              = 3
  ITEM_BYTES         = 10
  FORMAT             = "F10.3"
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  UNIT               = "km/s"
  DESCRIPTION        = "Velocity Vector in the spherical JUNO_JSS
                        (Jupiter-deSpun-Sun) frame in units of km/s.
                        Three components are provided:
                           [V_r, V_theta, V_phi]"
/* RJWcsv, V_JSSRTP_KMPS, d, 1, 3 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "V_JSSRTP_SIGMA_KMPS"
  FIELD_NUMBER       = 22
  BYTES              = 34
  ITEMS              = 3
```

```
  ITEM_BYTES         = 10
  FORMAT             = "F10.3"
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  UNIT               = "km/s"
  DESCRIPTION        = "Velocity Vector uncertainty in the spherical JUNO_JSS
                        (Jupiter-deSpun-Sun) frame in units of km/s.
                        Three components are provided:
                           [V_sigma_r, V_sigma_theta, V_sigma_phi]"
/* RJWcsv, V_JSSRTP_SIGMA_KMPS, d, 1, 3 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "PRESSURE_PA"
  FIELD_NUMBER       = 23
  BYTES              = 10
  FORMAT             = "E10.3"
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  UNIT               = "Pa"
  DESCRIPTION        = "Isotropic pressure in units of Pascals."
/* RJWcsv, PRESSURE_PA, d, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "PRESSURE_SIGMA_PA"
  FIELD_NUMBER       = 24
  BYTES              = 10
  FORMAT             = "E10.3"
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  DESCRIPTION        = "Isotropic pressure uncertainty in units of Pascals."
/* RJWcsv, PRESSURE_SIGMA_PA, d, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "TEMP_EV"
  FIELD_NUMBER       = 25
  BYTES              = 10
  FORMAT             = "E10.3"
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  UNIT               = "eV"
  DESCRIPTION        = "Isotropic temperature in units of eV."
/* RJWcsv, TEMP_EV, d, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "TEMP_SIGMA_EV"
  FIELD_NUMBER       = 26
  BYTES              = 10
  FORMAT             = "E10.3"
  DATA_TYPE          = "ASCII_REAL"
  MISSING_CONSTANT   = -9.999E+09
  UNIT               = "eV"
  DESCRIPTION        = "Isotropic temperature uncertainty in units of eV."
/* RJWcsv, TEMP_SIGMA_EV, d, 1, 1 */
END_OBJECT           = FIELD

OBJECT               = FIELD
  NAME               = "QUALITY_FLAG"
  FIELD_NUMBER       = 27
  BYTES              = 3
```

```
  FORMAT             = "I3"
  DATA_TYPE          = "ASCII_INTEGER"
  VALID_MINIMUM      =   0
  VALID_MAXIMUM      = 254
  MISSING_CONSTANT   = 255
  DESCRIPTION        = "Moments Quality Flag.
                         To be determined for future versions:
                         255 = Unknown = MISSING_CONSTANT
                         Currently this object is all values of 255."
/* RJWcsv, QUALITY_FLAG, I, 1, 1 */
END_OBJECT           = FIELD
```