INTEGRAL Science Data Centre

Introduction to the *INTEGRAL* Data Analysis

Reference	:	ISDC/OSA-INTRO
Issue	:	10.2
Date	:	Sep. 2020

INTEGRAL Science Data Centre
Chemin d'Écogia 16
CH–1290 Versoix
Switzerland

http://isdc.unige.ch

Authors and Approvals

ISDC	Introduction to the *INTEGRAL* Data Analysis	
Sep. 2020	10.2	

Prepared by : M. Chernyakova
A. Frankowski, L. Pavan, M. Türler

Agreed by : C. Ferrigno ....

Approved by : T. J.-L. Courvoisier, C. Ferrigno ....

Document Status Sheet

ISDC	Introduction to the *INTEGRAL* Data Analysis	
2 April 2003	1.0	First Release.
18 July 2003	2.0	Second Release.
8 December 2003	3.0	Third Release.
19 July 2004	4.0	Fourth Release.
6 December 2004	4.2	Update of Fourth Release.
29 June 2005	5.0	Fifth Release.
24 November 2006	6.0	Update of Fifth Release.
30 May 2012	10.0	Global review for OSA10.
21 August 2015	10.1	Update for change of INTEGRAL orbit.
20 September 2020	11.1	OSA 11 issues.
15 SEP 2020	Printed	

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AD	Architectural Design
AGB	Asymptotic Giant Branch
AGN	Active Galactic Nuclei
ARF	Ancillary Response Files
BGO	Bismuth Germanate
CdTe	Cadmium-Telluride
CsI	Caesium-Iodide
COP	Centre of dither Pattern
DOL	Data Object Locator
DS	Data Structure
ESA	European Space Agency
FAQ	Frequently Asked Questions
FITS	Flexible Image Transport System
FWHM	Full Width at Half Maximum
GRB	Gamma Ray Burst
GTI	Good Time Interval
ID	Identification
IDAS	*INTEGRAL* Data Analysis System
IC	Instrument Characteristics
ISOC	*INTEGRAL* Science Operations Centre
ISDC	*INTEGRAL* Science Data Centre
ISM	Interstellar Medium
IT	Instrument Team
MJD	Modified Julian Date
MOC	Mission Operations Centre
NASA	National Aeronautics and Space Administration
NRT	Near Real Time
OG	Observation Group
OMC	Optical Monitoring Camera
QLA	Quick-Look Analysis
OSA	Off-line Scientific Analysis
RMF	Redistribution Matrix Files
SA	Standard Analysis
S/C	Spacecraft
ScW	Science Window
ScWID	Science Window IDentifier
SWG	Science Window Group
TM	Telemetry
TOO	Target of Opportunity
WR star	Wolf-Rayet star

Glossary of Terms

- *ISDC system:* the complete ground software system devoted to the processing of the *INTEGRAL* data and running at the ISDC. It includes contributions from the ISDC and from the *INTEGRAL* instrument teams.
- *Science Window (ScW):* For the operations, ISDC defines atomic bits of *INTEGRAL* operations as either a pointing or a slew, and calls them ScWs. A set of data produced during a ScW is a basic piece of *INTEGRAL* data in the ISDC system.
- *Observation:* Any group of ScW used in the data analysis. The *observation* defined from ISOC in relation with the proposal is only one example of possible *ISDC observations.* Other combinations of Science Windows, *i.e.,* of observations, are used for example for the Quick-Look Analysis, or for Off-Line Scientific Analysis.
- *Pointing:* Period during which the spacecraft axis pointing direction remains stable. Because of the *INTEGRAL* dithering strategy, the nominal pointing duration is of order of 20 minutes.
- *Slew:* Period during which the spacecraft is manoeuvred from one stable position to another, i.e., from one pointing to another.

1 Introduction

The *INTEGRAL* Off-line Scientific Analysis (OSA) package is what you need to reduce and analyze data collected by the *INTEGRAL* satellite.

OSA is broken down into the following components:

1. Off-line Scientific Analysis Software (OSA_SW)
2. Instrument Characteristics (OSA_IC)
3. 'High-energy' Catalog (OSA_CAT)
4. Test Observations, for example to check the correct installation of the *INTEGRAL* Data Analysis System (OSA_TESTDATA)

The analysis is achieved by four independent instrument specific analysis chains (scripts). With the help of graphical user interfaces you may alter the analysis parameters to customize the analysis runs to your personal needs.

If *INTEGRAL* data analysis is new for you, a set of documents has been compiled to help you find your way:

1. **ISDC/OSA-INTRO**, the 'Introduction' to OSA (i.e., this document)
   Here you find all kinds of information on *INTEGRAL* and the data analysis. This document contains the following chapters:
   (a) **Introduction**: The current chapter.
   (b) ***INTEGRAL* and its Operations**: gives a general description of the *INTEGRAL* satellite and its instruments.
   (c) ***INTEGRAL* Data**: is devoted to the *INTEGRAL* data, the ISDC data model and the storage of *INTEGRAL* data.
   (d) **Software Tools**: gives an overview of the generic software tools included in the release.
   (e) **Setting up the environment**: instructs you how to set up the working environment before running the software.
   (f) **Data Analysis**: provides an overview of the methodology of the *INTEGRAL* data.
2. **ISDC/OSA-INST-GUIDE**, the 'Installation Guide' for all OSA components
   Here you will find the instructions to install the OSA components together with the system requirements.
3. **ISDC/OSA-UM-⟨instrument_name⟩**, the instrument specific 'Analysis User Manual'
   Here you find a brief description of the instruments aboard *INTEGRAL*, a cookbook with examples on how to run the analysis, and the descriptions of the scientific algorithms applied.
4. **ISDC/OSA-SCI-VAL-⟨instrument_name⟩**, the instrument specific 'Scientific Validation Reports'

The schematic overview of the documents related to OSA is shown in Figure 1. All these documents can be found at:
http://www.isdc.unige.ch/integral/analysis

For questions or remarks concerning the *INTEGRAL* Data Analysis System a list of "Frequently Asked Questions" is available at the following URL:
http://www.isdc.unige.ch/integral/analysis#FAQ

Alternatively, you may send an e-mail to the *INTEGRAL* help-desk:
inthelp@rssd.esa.int

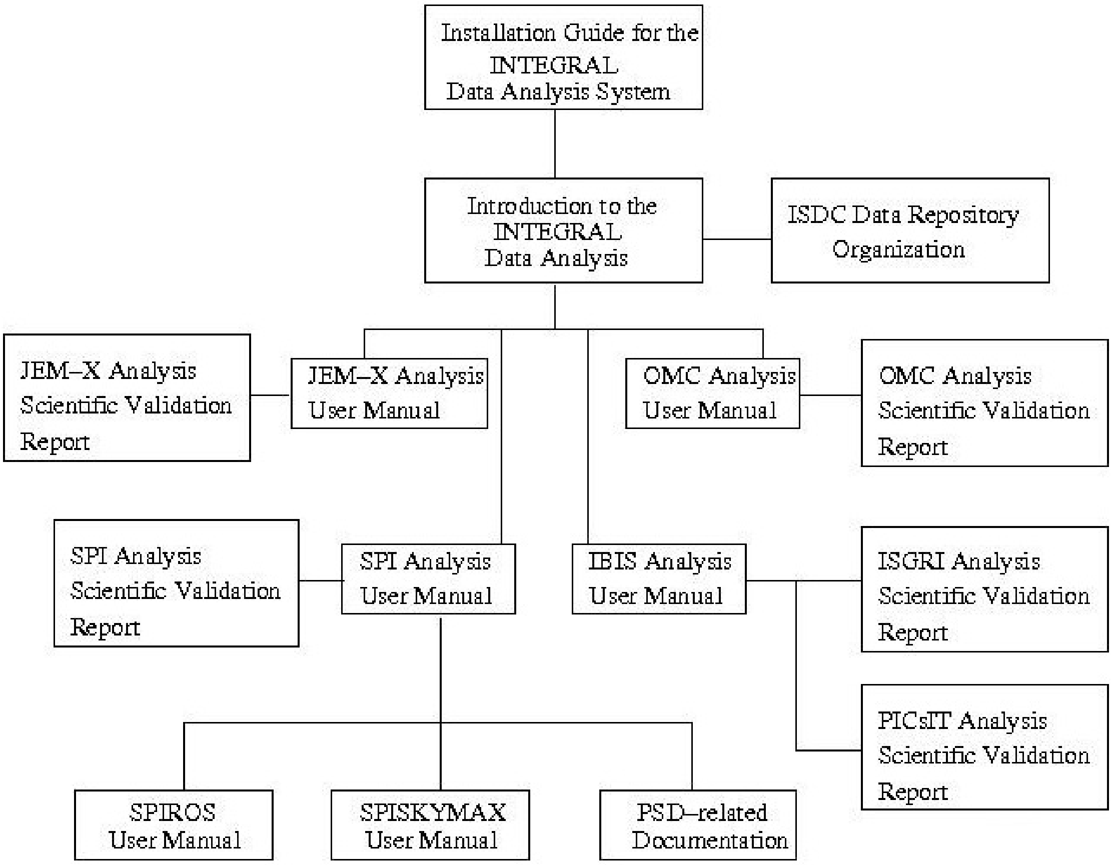
Installation Guide for the INTEGRAL Data Analysis System
Introduction to the INTEGRAL Data Analysis
ISDC Data Repository Organization
JEM–X Analysis Scientific Validation Report
JEM–X Analysis User Manual
OMC Analysis User Manual
OMC Analysis Scientific Validation Report
SPI Analysis Scientific Validation Report
SPI Analysis User Manual
IBIS Analysis User Manual
ISGRI Analysis Scientific Validation Report
PICsIT Analysis Scientific Validation Report
SPIROS User Manual
SPISKYMAX User Manual
PSD–related Documentation

Figure 1: Overview of the *INTEGRAL* analysis related documents.

2 *INTEGRAL* and its Operations

2.1 General description

The **INTE**rnational **G**amma-**R**ay **A**strophysics **L**aboratory (*INTEGRAL*) is a 15 keV–10 MeV gamma-ray observatory mission with concurrent source monitoring in the X-rays (3–35 keV) and in the optical range (V band, 500–600 nm). It is the second medium size satellite of the ESA long term scientific plan Horizon 2000. *INTEGRAL* is an observatory type facility with most of the observing time open for the astronomical community. The mission is conceived as an observatory led by ESA with contributions from the Russian Space Agency (PROTON launcher) and NASA (Deep Space Network ground station). The overall view of the *INTEGRAL* is given in Figure 2.

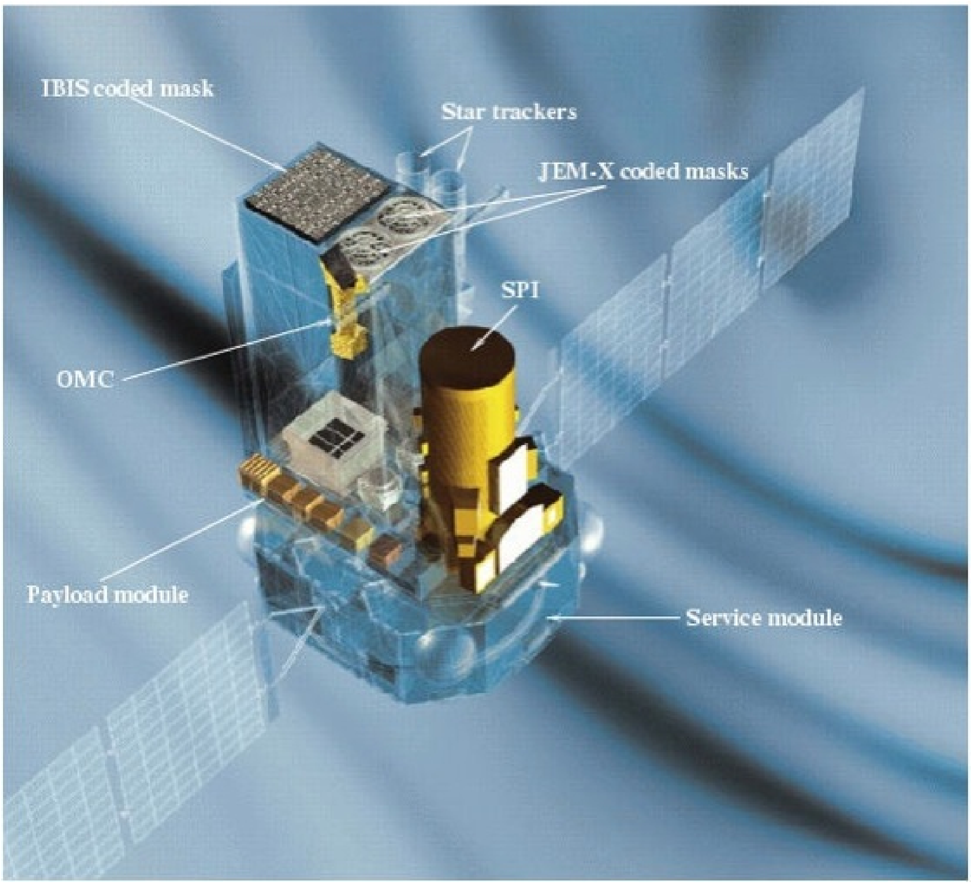

Figure 2: The *INTEGRAL* spacecraft with the payload module on the top of the service module. The coded masks for IBIS and JEM-X are located 3.2 m above the detection planes. IBIS and JEM- X detectors are inside the payload module structure. The overall dimensions of the spacecraft (excluding solar arrays which span 16 m) are 4 m × 6 m (w×h). The Sun is illuminating the payload module on the IBIS side, leaving SPI in the shadow.

The scientific goals of *INTEGRAL* are attained by high resolution spectroscopy with fine imaging and accurate positioning of celestial sources of gamma-ray emission. High resolution spectroscopy over the entire energy range permits spectral features to be uniquely identified and line profiles to be determined for physical studies of the source region. The fine imaging capability of *INTEGRAL* within a large field of view allows an accurate localization and hence identification of the gamma-ray emitting objects with their counterparts at other wavelengths, enables extended regions to be distinguished from point sources and provides considerable serendipitous science which is very important for an observatory-class mission. In summary the *INTEGRAL* scientific topics address:

- compact objects: white dwarfs, neutron stars, black hole candidates, high energy transients and Gamma Ray Bursts (GRBs);
- extragalactic astronomy: AGNs, Seyferts, blazars, galaxies and clusters, cosmic diffuse background;
- stellar nucleosynthesis: hydrostatic nucleosynthesis (AGB and WR stars), explosive nucleosynthesis (supernovae and novae);
- Galactic structure: mapping of continuum and line emission, ISM, cosmic-ray distribution;
- Galactic Centre: cloud complex regions, mapping of continuum and line emission, ISM, cosmic-ray distribution;
- particle processes and acceleration: trans-relativistic pair plasmas, beams, jets;
- identification of high energy sources: unidentified gamma ray objects as a class;
- unexpected discoveries.

2.2 On-Board Instruments

The *INTEGRAL* payload consists of two main gamma-ray instruments: the Spectrometer, SPI, and the Imager, IBIS. Each of them has both spectral and angular resolution, but they are differently optimized in order to complement each other and to achieve overall excellent performance. These instruments are supported by two monitor instruments which provide complementary observations in the X-ray and optical energy bands: the X-Ray Monitor, JEM-X, and the Optical Monitoring Camera, OMC. The Spectrometer, Imager and JEM-X share a common principle of operation: they are all coded-mask telescopes. More details are given in Tables 1 and 2. The continuum sensitivities of the *INTEGRAL* instruments are shown in Figure 4. The detailed description of each instrument is given in the corresponding User Manuals [1], [2],[3],[4].

The coordinate system incident to the *INTEGRAL* is an X, Y, Z orthogonal coordinate system with the origin at the center of the separation plane between spacecraft and launch adaptor, see Figure 3. The X-axis is perpendicular to this spacecraft/launcher separation plane, pointing positively from the separation plane towards the spacecraft (i.e. the X-axis is the pointing direction). The Z-axis is orthogonal to the solar array surface, pointing positively to the sun, i.e. the sun is illuminating the payload module on the IBIS side, while leaving SPI in the shadow. The Y -axis completes the coordinate system.

Table 1: Key performance parameters of SPI & IBIS Instruments.

	SPI	IBIS
Energy range	20 keV 8 MeV	15 keV 10 MeV
Detector	19 Ge detectors cooled to 85 K	16384 CdTe detectors (ISGRI), 4096 CsI detectors (PICsIT)
Detector area (cm2)	500	2600 (CdTe), 3100 (CsI)
Spectral resolution (FWHM)	2.3 keV @ 1.3 MeV	9 keV @ 100 keV
Field of view (fully coded)	16° (corner to corner)	9° × 9°
Angular resolution (FWHM)	2.5° (point source)	$12'$
Source location (radius)	$< 1.3^{\circ}$ (depending on the source strength)	$< 1'$ (for 10σ source)
Absolute timing accuracy (3σ)	129 μs	92μs

2.2.1 IBIS

Principal Investigator:

Table 2: Key performance parameters of JEM-X & OMC Instruments.

	JEM-X	OMC
Energy range	3 keV 35 keV	500 nm 600 nm
Detector	Microstrip Xe/CH4 -gas detector (1.5 bar)	CCD + V-filter
Detector area (cm2)	2 × 500	CCD: (2061×1056) pixels Imaging area: (1024 ×1024) pixels
Spectral resolution (FWHM)	1.2 keV @ 10 keV	
Field of view (fully coded)	4.8°	5° ×5°
Angular resolution (FWHM)	3′	23″
10σ source location (radius)	<30″	2″
Absolute timing accuracy (3σ)	122 μs	>1 s

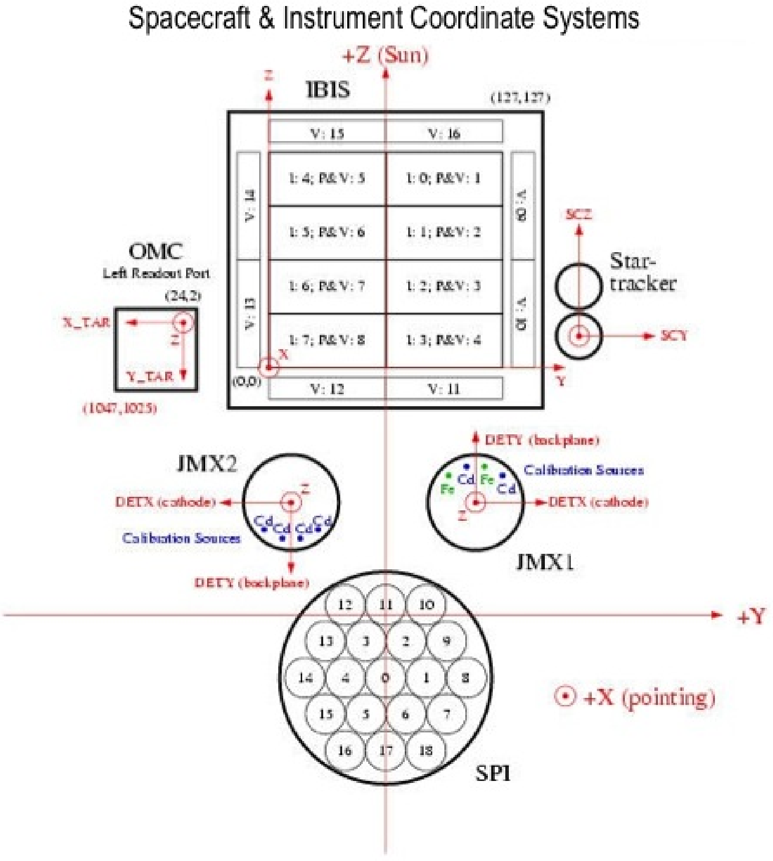

Figure 3: Spacecraft & Instrument coordinate systems.

- Pietro Ubertini, INAF/IAPS, Rome, Italy

Co-Principal Investigators:

- Philippe Laurent (formerly Francois Lebrun), CEA-Saclay, France
- Guido Di Cocco, INAF/IASF-Bologna, Italy

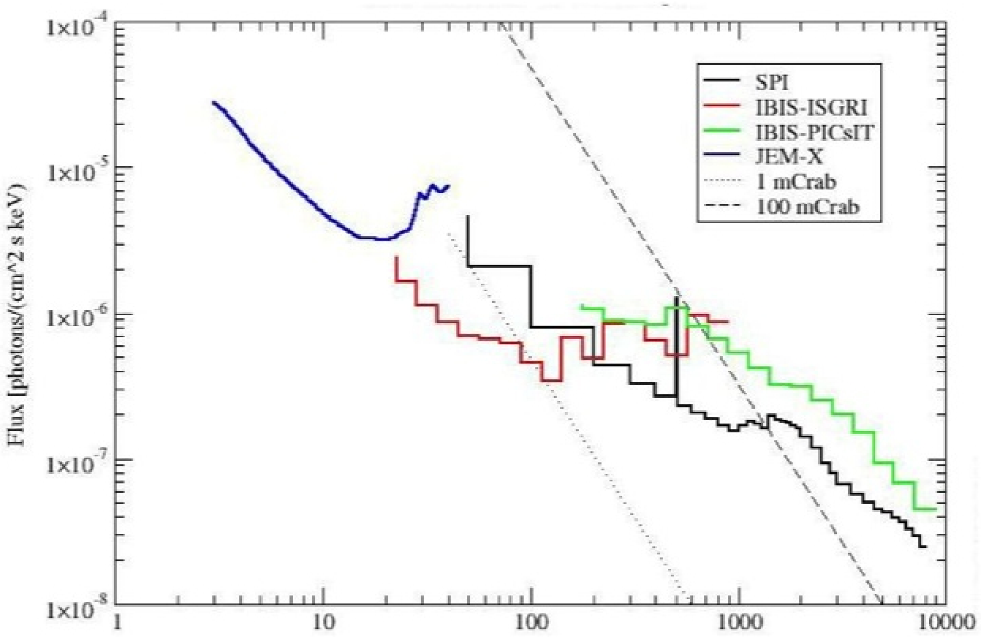

Figure 4: The *INTEGRAL* continuum sensitivities. Time of observation is 10^6 s, the detection threshold is 3σ.

with collaborating scientific institutes in Italy (INAF/IAPS Rome, INAF/IASF-Bologna, INAF/IASF-Palermo), France (CEA Saclay), Norway (U Bergen), Germany (IAA-Tübingen), Spain (University of Valencia), USA (NASA/MSFC Huntsville), Poland (Space Research Centre, Warsaw) and UK (University of Southampton).

The Imager IBIS provides diagnostic capabilities of fine imaging (12 arcmin FWHM), source identification and spectral sensitivity to both continuum and broad lines over a broad (15 keV–10 MeV) energy range. The Imager observes, simultaneously with the other instruments of *INTEGRAL* , celestial objects of all classes ranging from the most compact galactic systems to extragalactic objects. A tungsten coded-aperture mask (located at 3.2 m above the detection plane) is optimized for high angular resolution. As diffraction is negligible at gamma-ray wavelengths, the angular resolution obtainable with a coded mask telescope is limited by the spatial resolution of the detector array.

The Imager design takes advantage of this by utilizing a detector with a large number of spatially resolved pixels implemented as physically distinct elements. The detector uses two planes: one 2600 cm^2 front layer of Cadmium-Telluride (CdTe) pixels, each (4×4×2) mm (width × depth × height); and a 3100 cm^2 layer of Caesium-Iodide (CsI) pixels, each (9×9×30) mm. The CdTe array (ISGRI) and the CsI array (PICsIT) are separated by 90 mm. The detector provides the wide energy range and high sensitivity continuum spectroscopy required for *INTEGRAL* . The two-layer structure allows the paths of the photons to be tracked in 3D, as they scatter and interact with more than one element. Events can be categorized and the signal to noise ratio improved by rejecting those which are unlikely to correspond to real (celestial) photons, e.g., towards the high end of the energy range. The aperture is restricted by a lead shielding tube and shielded in all other directions by an active Bismuth Germanate (BGO) scintillator veto system.

The dependence of the IBIS effective area on energy is given in Figure 5.

If you want to know more about IBIS we recommend that you start with [3], where we describe the limits of validity of different OSA versions.

2.2.2 SPI

Co-Principal Investigators:

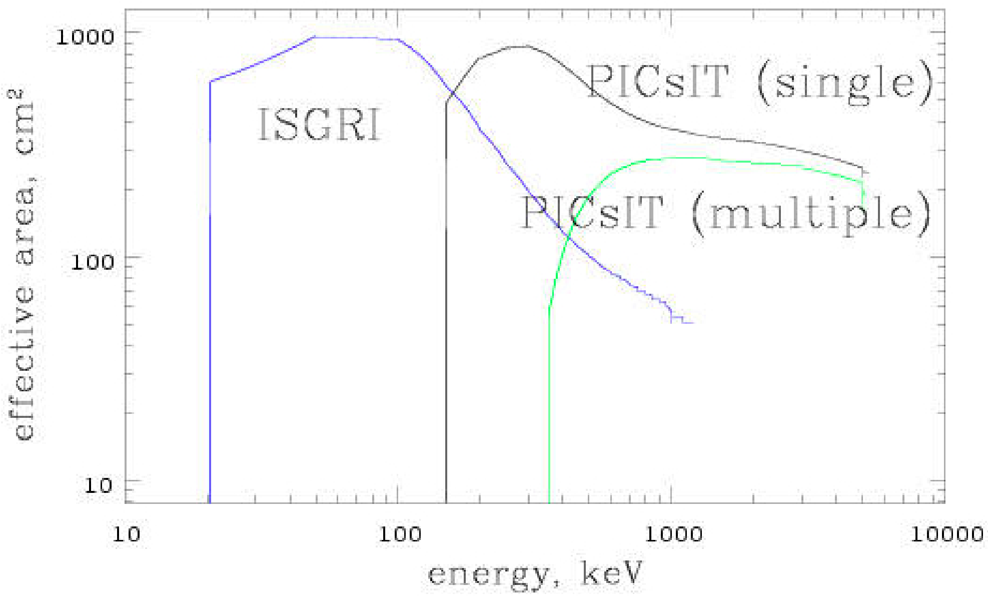

Figure 5: Energy dependence of the IBIS effective area.

- Jean-Pierre Roques, CESR Toulouse, France
- Jochen Greiner (formerly Roland Diel), MPE Garching, Germany

with collaborating scientific institutes in France (CESR Tolouse, CEA Saclay, CNES Toulouse), Germany (MPE Garching), Italy (INAF/IASF-Milano), Spain (University Valencia), Belgium (University Louvain), United Kingdom (University Birmingham), USA (UC San Diego, LBL Berkeley, NASA/GSFC Greenbelt). The spectrometer SPI performs spectral analysis of gamma-ray point sources and extended regions in the 20 keV–8 MeV energy range with an energy resolution of 2.2 keV (FWHM) at 1.33 MeV. This is accomplished with an array of 19 hexagonal high purity Germanium detectors cooled by a Stirling cooler system to an operating temperature of 85 K. A hexagonal coded aperture mask is located 1.7 m above the detection plane in order to image large regions of the sky (fully coded field of view is 16 degrees) with an angular resolution of 2 degrees. In order to reduce background radiation, the detector assembly is shielded by a veto (anticoincidence, ACS) system which extends around the bottom and sides of the detector almost completely up to the coded mask. The aperture (and hence contribution by cosmic diffuse radiation) is limited to 30 degrees. A plastic veto is provided below the mask to further reduce the 511 keV background.

If you want to know more about SPI, we recommend that you start with the SPI Analysis User Manual [2].

2.2.3 JEM-X

Principal Investigator:

- Søren Brandt, DTU-Space, Copenhagen, Denmark

with collaborating scientific institutes in Denmark (DSRI Lyngby), Finland (Metorex Espoo, University Helsinki), Spain (University Valencia, INTA Madrid), Italy (INAF/IASF-Frascati, University Ferrara, INAF/IASF-Palermo, INAF/IASF-Bologna), USA (NASA/GSFC Greenbelt), Sweden (Observatory Stockholm), United Kingdom (University Cambridge), Poland (Copernicus Center Warsaw, Space Research Center Warsaw), Russia (IKI Moscow).

The Joint European X-Ray Monitor JEM-X supplements the main *INTEGRAL* instruments and plays a crucial role in the detection and identification of the gamma-ray sources and in the analysis and scientific interpretation of *INTEGRAL* gamma-ray data. JEM-X makes observations simultaneously with the main gamma-ray instruments, albeit with a narrower field of view, and provides images with arcminute angular resolution in the 3–35 keV energy band. The baseline photon detection system consists of two identical high pressure imaging microstrip gas chambers (1.5 bar, 90% Xenon + 10% Methane). Each detector unit views the sky through its coded aperture mask located at a distance of approximately 3.2 m above the detection plane.

Due to unforeseen problems with eroding anodes, detected after launch, the high voltage inside the two JEM-X detectors has been lowered, reducing the gain by a factor of 3 for JEM-X 1 and a factor of 2 for JEM-X 2. As additional safeguard measure, only JEM-X 1 was operated for normal observations for most of the mission lifetime. However, at the time of writing both units are operative.

The dependence of the JEM-X effective area on energy under the current conditions is given in Figure 6.

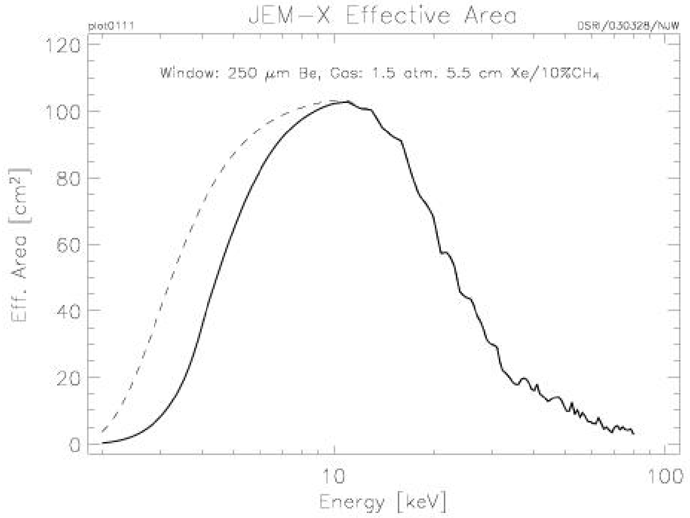

Figure 6: JEM-X effective area with the mask taken into account. The dashed line shows the effective area before the high voltage reduction (see text) and the full curve shows the efficiency when taking into account the effect of the electronic low-signal cutoff.

If you want to know more about JEM-X we recommend that you start with the JEM-X Analysis User Manual [1].

2.2.4 OMC

Principal Investigator:

- Miguel Mas-Hesse, LAEFF-INTA, Madrid

with collaborating scientific institutes in Spain (INTA/LAEFF Madrid, University Valencia, University Barcelona), Ireland (UC Dublin), Belgium (University Liege), United Kingdom (MSSL Dorking), Czech Republic (Astroph. Institute).

The Optical Monitoring Camera OMC consists of a passively cooled CCD (2055×1056 pixels, imaging area: 1024×1024 pixels) working in frame transfer mode. The CCD is located in the focal plane of a 50 mm (diameter) lens including a Johnson V-filter to cover the 500–850 nm wavelength range. The OMC is mounted close to the top of the payload module structure.

The OMC observes the optical emission from the prime targets of the *INTEGRAL* main gamma-ray instruments and the X-Ray Monitor JEM-X. The OMC offers the first opportunity to make long observations in the optical band simultaneously with those at X-rays and gamma-rays. This capability provides diagnostic information on the nature and the physics of the sources over a broad wavelength range. Multiwavelength observations are particularly important in high-energy astrophysics where the source variability is typically rapid. If you want to know more about OMC we recommend you to start with the OMC Analysis User Manual [4].

2.2.5 IREM

Developed and manufactured by:

- OERLIKON-CONTRAVES SPACE in cooperation with the PAUL SCHERRER INSTITUTE (PSI) under a development contract of the EUROPEAN SPACE AGENCY (ESA)

The *INTEGRAL* Radiation Environment Monitor (IREM) performs a wide range of radiation monitoring functions in-orbit and downloads the results via the *INTEGRAL* spacecraft telemetry to the ground.

The IREM data are part of the spacecraft "housekeeping" stream and, as such, are made available to all users via the ISDC.

The IREM unit measures energetic electrons, protons, and heavy ions, as well as the total accumulated dose, encountered during the mission. This information is made available in real time to the other experiments on-board the *INTEGRAL* satellite and, in due course, implemented as a part of the overall Standard Radiation Environment Monitors (SREM) measurement data base at the Paul Scherrer Institute, Switzerland, together with data from SREM units on other missions. Up to now, the SREMs have allowed to accumulate valuable new results concerning, for example, flux directionality, electron belt dynamism, and seasonal changes.

2.3 Strategies of Scientific Observations

Three types of scientific observation strategies are distinguished for *INTEGRAL* :

- dithering,
- scanning,
- staring.

The method of dithering is employed to suppress systematic effects on spatial and temporal background variations in the spectrometer and imager detectors. A controlled and systematic spacecraft dithering (Raster Scan) manoeuvre capability is required. Such manoeuvres consist of several off-pointings of the spacecraft pointing axis from the target. Two dithering patterns have been defined, which are applied in nominal science operation mode. These patterns are called hexagonal pattern and rectangular pattern (5×5 pattern), respectively. A schematic view of dithering patterns and instrument fully coded field of view is given in Figure 7.

The hexagonal pattern is used for a single known gamma-ray point source and covers sequentially the six edges of a hexagon and the nominal source location at individual steps of 2°, as shown in Figure 8. Observers are generally discouraged from using this mode, because it seriously compromises the imaging capabilities of SPI and also of IBIS, rendering the data useless for use in large mosaics. The use of the hexagonal mode must be well motivated and is accepted if the scientific goals require continuous monitoring of the main target by JEM-X. Such a strategy would however be at the expense of SPI data quality if there are even a few sources in the FOV. This observation mode was altered in November 2, 2005 to allow for a wandering Centre of dither Pattern (COP) offset to the hexagonal dither pattern. This COP pattern consists of 2×6 points that define two hexagons, centered around the original centre point of the dither pattern.

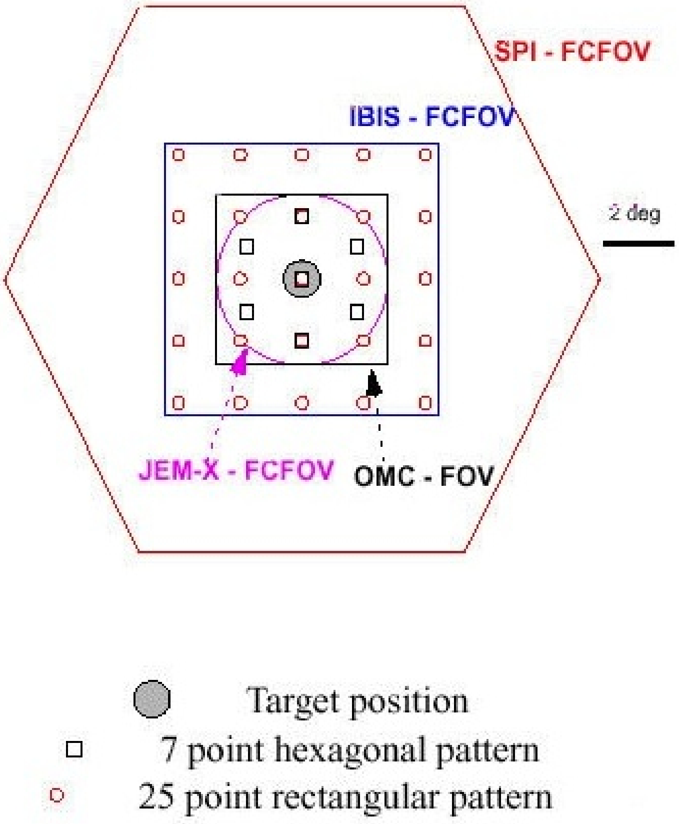

Figure 7: Schematic view of dithering patterns and instrument fully coded fields of view (FCFOV).

The rectangular 5×5 pattern is the default mode and is used for multiple point sources in the field of view, for sources with unknown locations, and for extended diffuse emissions. The pattern covers a 5°× 5° pointing pattern around the target with a step size of 2° .

During the first two years of operations, this mode consisted simply of a square pattern centred on the nominal target position. In this implementation, one pointing was with the source on-axis, and 24 other pointings with the source off-axis, each separated by 2.17° arranged on a rectangular grid. The roll angle between pointings was always 0°.

Afterwards, the pattern was optimised to reduce systematic effects in the IBIS images. This implies that for observations requiring several dither cycles, an offset between the centre of each dither cycle is introduced. This ensures that no pointing attitude is repeated over the course of the observation. Hence, the COP moves around in a pre-defined manner during the course of an observation. The COP pattern is parallel to the original 5×5 dither and consists of 7×7 points centred on the target, with a step size of 0.3°. Thus, the whole COP pattern fits within the inner 3×3 points of the original dither. The 49 points in the COP pattern allow for an observation time of 2.2 Ms without repetition of a given pointing.

In addition to the moving COP, to further reduce systematic effects in deep mosaics, since June 2005 the orientation of the 5×5 pattern is set such that the axis of the dither pattern is rotated by $11.3° = \arctan(1/5)$ with respect to the instrument axes. As the instrument axes depend on the relative position of the Sun, the exact dither pattern pointing directions depend on the time of execution of the observation.

The most recent optimization to reduce systematic noise in mosaics involves a stepping in roll angle, and was implemented for the first time at the end of November 2007 in revolution 624. With this strategy, the roll angle for an observation with N repetitions of the 5×5 pattern spans the range from $+3°$ to $-3°$, in steps of $d\theta = 6/(N-1)$.

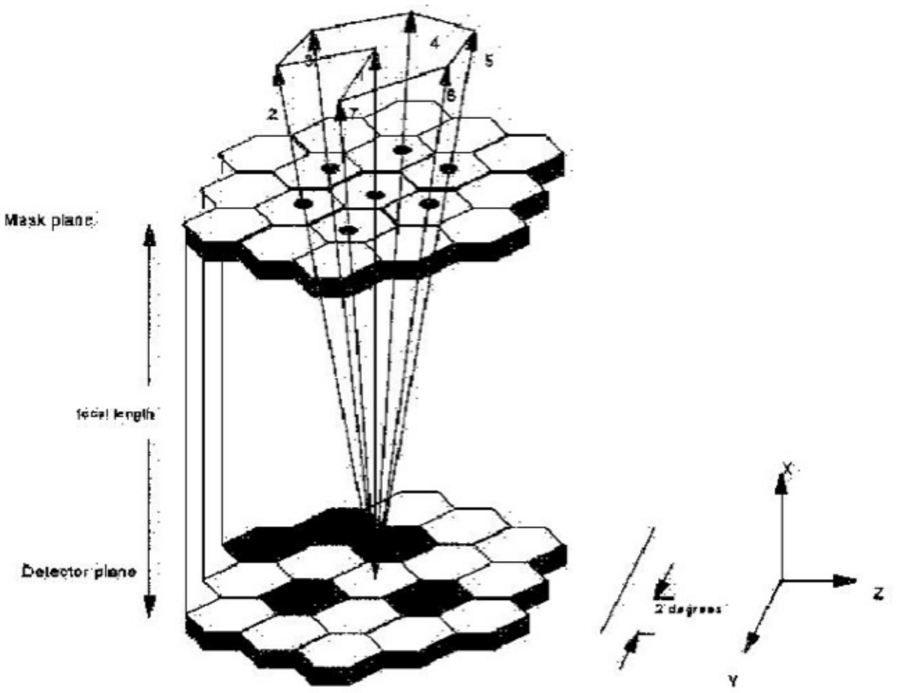

Figure 8: Hexagonal dithering pattern.

Scanning strategies are used to cover large parts of the sky for monitoring purposes or for detecting diffuse emission in what is called "custom modes". Among these, the "Galactic Plane Scan" has been codified as follows. At weekly intervals, a saw-tooth-path scan inclined at 21° with regard to the galactic equator is employed, see Figure 9. Each scan consists of a series of individual exposures of 965 s each. The individual exposures are separated by 6° along the scan path.

Staring observations are used for calibration purposes or in circumstances that require staring on-axis observations of a source, such as in studies of time variability or QPOs. The use of this mode is strongly discouraged, because it very seriously compromises the imaging capabilities of SPI, IBIS, and JEM-X, rendering the data useless for use in large mosaics, and in particular for the study of high energy emission from celestial sources using SPI.

2.4 Operational Orbit

The original *INTEGRAL* orbit is a highly excentric one, inclined by 51.6° and with a perigee height of ∼9000 km and an apogee height of 152 600 km. The period of the *INTEGRAL* orbit is 3 sidereal days, so that the perigee occurs always above the same geographical point on Earth. Since one sidereal day lasts 23h 56m 4s, the local time of the perigee changes by ~ -12 minutes on every revolution. While the perigee height and the inclination have changed a lot over the years, the period remained constant until 2015.

In early 2015, ESA performed a series of manoeuvres towards a safe disposal of INTEGRAL in February 2029, via natural re-entry in the atmosphere. These operations took place from 13 January (end of rev. 1495) to 12 February 2015 (end of rev. 1506). The resulting orbit is no longer of 3 sidereal days, but has a repeating pattern of 3 revolutions in 8 sidereal days. The orbit period is thus of roughly 2 days and 16 hours. The loss of science time due to a higher fraction of perigee passages does not exceed 5%, and the seasonal visibility does not change significantly either.

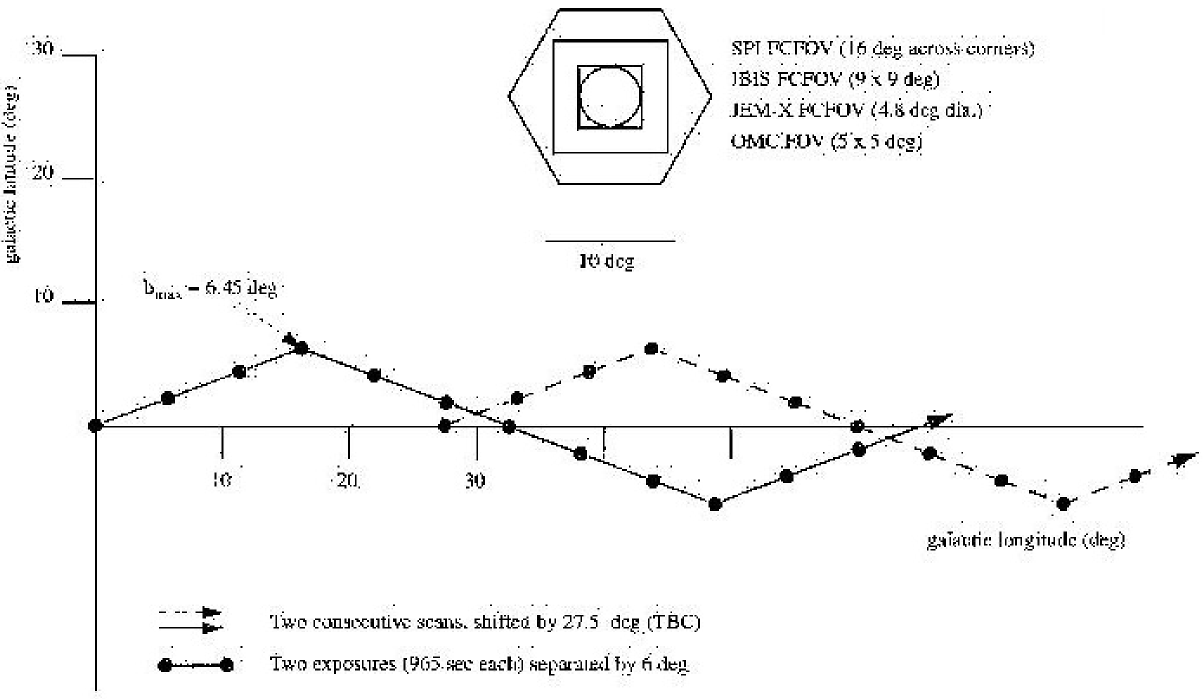
30
20
10
galactic latitude (deg)
b_max = 6.45 deg
SPI FCFOV (16 deg across corners)
IBIS FCFOV (9 x 9 deg)
JEM-X FCFOV (4.8 deg dia.)
OMC FOV (5 x 5 deg)
10 deg
10
20
30
galactic longitude (deg)
Two consecutive scans, shifted by 27.5 deg (TBC)
Two exposures (965 sec each) separated by 6 deg

Figure 9: Galactic Plan Scan.

3 *INTEGRAL* data

An *INTEGRAL* revolution of three days (about 72 hours) results in a telemetry volume of approximately 2 Gb. This telemetry stream is processed and analyzed at the ISDC and will result in about 15 Gb of uncompressed data products per revolution. These data are stored in FITS files and consist of:

- raw data (4.5 Gb), which are reformatted data with the same information content as the telemetry sent by the spacecraft,
- prepared data (7.2 Gb) including additional timing information,
- corrected data (2.7 Gb) including gain corrected event energy.

All these data are distributed in a compressed format (∼9 Gb per revolution) to the observers. Proposers should be therefore aware that they need more than 10 Gb of disk-space to store *INTEGRAL* observations that last about one revolution.

3.1 How to get *INTEGRAL* data

All the data related to your proposal will be available for a download through FTP or `rsync`. These files however will not include the instrument calibration (IC) files, which you should download from the ISDC public release web-page:

http://www.isdc.unige.ch/integral/analysis#Software

When the data become public you can download them from the archive:

http://www.isdc.unige.ch/integral/archive#Browse

Examples on the downloading data from the archive can be found at the Cookbook Sections of [1], [2],[3],[4].

3.2 Science Windows, Science Window Groups, & Observation Groups

Because of the dithering of the *INTEGRAL* satellite (see Section 2.3), an observation is made of many pointings of about 30 minutes separated by slews. Each pointing and slew, or a part of them if they are too long, is called a “Science Window”.

Usually, the Observation consists of several Science Windows. For each Science Window, a few hundred data structures are grouped together in a compound object called a “Science Window Group” containing all the data belonging to this Science Window. At first, it consists of raw and prepared science data and housekeeping data. Housekeeping (HK) data consist of technical information related to the instruments, as for instance instrumental temperatures or voltages. During the Science Analysis of the data newly created files are attached to the group. Since an observation consists of many pointings, an “Observation Group” is defined as being a collection of many science window groups. An Observation Group contains all the data belonging to a given observation.

3.3 Hierarchical Data Storage

3.3.1 ISDC Data Model

The relationships among observations, pointings, and pointing data set components imply a natural hierarchy to the *INTEGRAL* data. This has led to the current ISDC Data Model which generalizes the concepts of pointings and observations into *Data Objects* and *Data Elements*.

The highest level of data organization within this model is the data object, or object. A data object is a compound structure consisting of one or more atomic structures known as data elements. Within a data object, the data elements are arranged in a hierarchy that often reflects the organization of a data set (e.g., science window, observation) where each data element contains a specific portion of the data set (e.g., spectra, event list, auxiliary information), and are distributed across a number of data files. For this reason, users may in practice often think of a data object as a data set.

The hierarchical relationship of data elements comprising a data object is analogous to a tree structure. Each tree node (or branch) corresponds to a parent element and its leaves are children elements of the parent element. The root of the tree structure, the only element without parents, is known as the *top-level element* of the object. The example of such a tree is given in Figure 10.

3.3.2 FITS Files and Data Object Locators (DOLs)

ISDC adopted the FITS format to handle and store the *INTEGRAL* data at each level of the data processing. The detailed information on FITS formats is available from the FITS Support Office at NASA/GSFC: http://fits.gsfc.nasa.gov

A FITS file consists of one or more HDU - an ASCII Header Unit followed by an optional Data Unit. The first HDU is called the ‘Primary HDU’. At ISDC, data analyses system primary HDU consists of only a header. In the following, we also use the term *data structure* along with HDU.

Each Header Unit consists of any number of 80-character keyword records which have the general form:

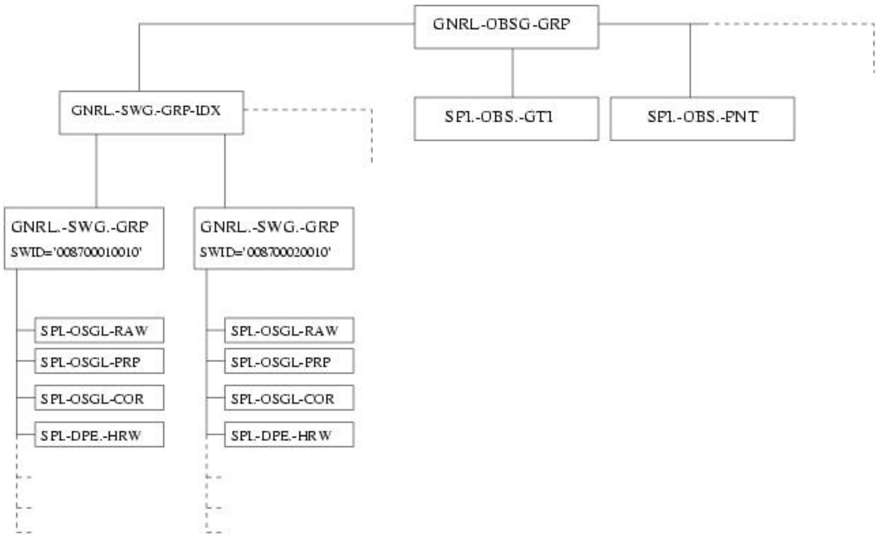

Figure 10: Schematic view of a SPI Observation Group.

KEYNAME = value / comment string

A keyword may also have a null or undefined value if there is no specified value. Any number of additional HDUs may follow the primary array; these additional HDUs are called FITS 'extensions'. There are currently 3 types of extensions defined by the FITS standard:

- Image Extension - a N -dimensional array of pixels
- ASCII Table Extension - rows and columns of data in ASCII character format
- Binary Table Extension - rows and columns of data in binary representation

To specify which data structure should be opened at run time the HDU name or number should be specified after the root file name. For example, 'myfile.fits[4]' opens the 4th extension, which corresponds to the 5th HDU in the file (note that the HDU numbering starts with 0), and 'myfile.fits[EVENTS]' opens the data structure with the name 'EVENTS' (as defined by the EXTNAME or HDUNAME keywords). If no HDU is specified, then the first extension is opened by default (i.e., this is equivalent to specifying 'myfile.fits[1]').

Such an ability to have an access to the desired data structure leads to the possibility of arranging a group with its members residing in separate FITS files. This allows data objects to be distributed across multiple data files, which has the following consequences:

- different data objects may simultaneously share the same data elements (e.g., two science windows may share the same orbit data),
- the construction of data object structures (e.g., science windows, observations) is not limited by the content of the individual data files,
- data objects may span multiple file systems and archives across the Internet,
- applications do not have to be concerned about what or how many data files are being accessed when they open a data object.

The particular existing data object is determined by specifying a Data Object Locator, or DOL. When an analysis script asks you to specify the DOL of the catalog you want to use in your analysis you should specify the path of the corresponding FITS file and the corresponding name or number (do not forget that numbering starts with 0!) of the data structure with the catalog in square brackets, e.g.

```
DOL of Reference Catalog []: ../../cat/hec/gnrl\_ref\_cat\_007.fits[1]
```

or

```
DOL of Reference Catalog []: ../../cat/hec/gnrl\_ref\_cat\_007.fits[GNRL-REFR-CAT]
```

omission of the extension is equivalent to '[1]'.

3.3.3 Groups and Indexes

The main objective of the usage of Data Objects is an easy and efficient access to the data, you should just specify the name of the desired Data Structure. However, there is an ambiguity as soon as two Data Elements have the same name. For these reasons the concept of Index was introduced. An Index is a FITS Group that, in addition to the usual stuff stored in FITS Groups, may contain columns describing the peculiarities of the Children. All Children in the Index Group have the same name. In ISDC Data Model only two cases are possible:

1. all direct members of the group have different names;
2. all direct members of the group have the same names. The group is therefore an Index.

In case 1, we may have to store several Data Elements that have the same name in the Object. However, and that is the reason it is stated "direct members", these Data Elements must be encapsulated in an Index.

Standard Groups are those whose direct members all have different names. Apart from the fact that direct members must all have different names, the only constraint is that the group name should follow the pattern:

XXXX-XXXX-GRP

All members of the index table are identical, but contain the additional information that helps to identify them. The name of Index Group follows the pattern:

XXXX-XXXX-XXX-IDX,

where XXXX-XXXX-XXX is the name of the children data structures.

Among other things, an Observation Group contains an Index, which members are Science Window Groups. An Observation Group must always contain this Index Group whose members are identical. This index allows the user to search effciently for Science Window Groups using the Science Window ID, time, and so on. The structure of an Observation Group is illustrated in Figure 10.

3.4 Data Repositories

As *INTEGRAL* data are gathered from the satellite, processed, and distributed to the community, they are stored in a number of data repositories. These repositories range from the Primary *INTEGRAL* Archive Repository residing at ISDC, to the data sets distributed to the General Observers.

Much of the software written for data analysis, as well as operations software designed to run at ISDC, rely on finding data in a pre-determined directory organization regardless of its origin. Thus, all data repositories must be compliant with a common set of organizational rules. Here we are speaking about the intended organization of all repositories used to store *INTEGRAL* data for use at ISDC or with ISDC-supplied analysis software. Such repositories are said to be ISDC-compliant repositories.

In October 2004, all public *INTEGRAL* data became available both in the older (revision 1) and in the revised format of the archive (rev. 2). In 2011 another revision of the archive was performed with the same structure as rev. 2. The new "Archive Revision 3" format is now the default on the Archive Browse WWW page at:

http://www.isdc.unige.ch/integral/archive#Browse

Current version of the Off-line Scientific Analysis (OSA) is working with any type of data.

The main differences between rev. 1 and the later data format are:

1. a simplified directory and file structure
2. the availability of event time stamps in Julian Dates (JD) instead of On-Board Time (OBT), with corrected energies, standard good time intervals (GTI) and deadtime in the Science Window datasets
3. the adoption of the standard OGIP format for event files making the timing analysis with FTOOLS much easier
4. the correction of all JD time stamps for the offsets between the OBT of each instrument (NB: the OBT values themselves are not corrected)
5. the implementation of some attitude handling optimization

The whole complicated structure of the revision 1 data is shown in Figure 11, and that of the revision 2-3 data in Figure 12. In the following subsections we explain the main branches step by step.

Note that when you start the analysis the only directory which differs from the shown structure is directory **obs**. This directory is filled in the process of the analysis.

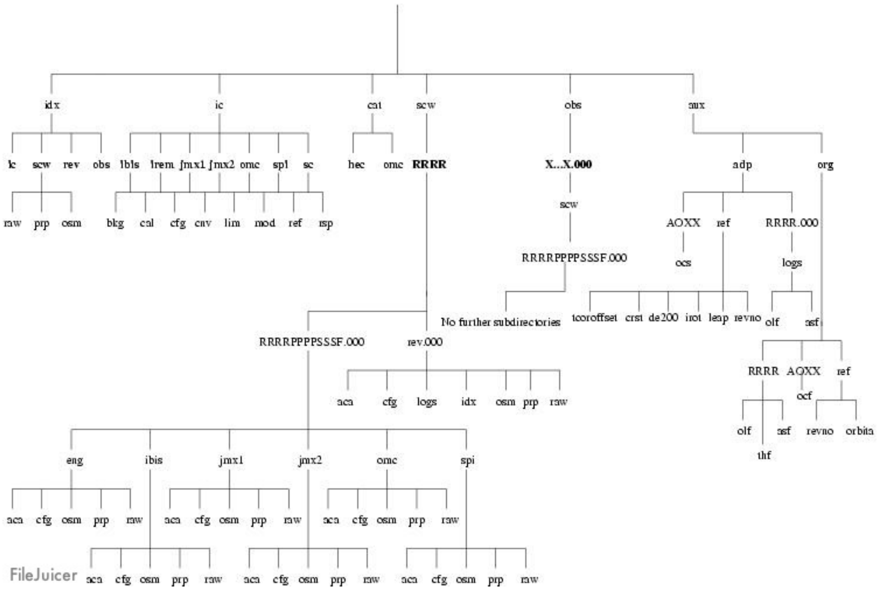

Figure 11: The complete ISDC repository structure of revision 1 data.

3.4.1 Processed data repository class

The processed data repository structure contains products created by the ISDC processing pipelines, along with their auxiliary and calibration data. This structure is used both for temporary storage of data products

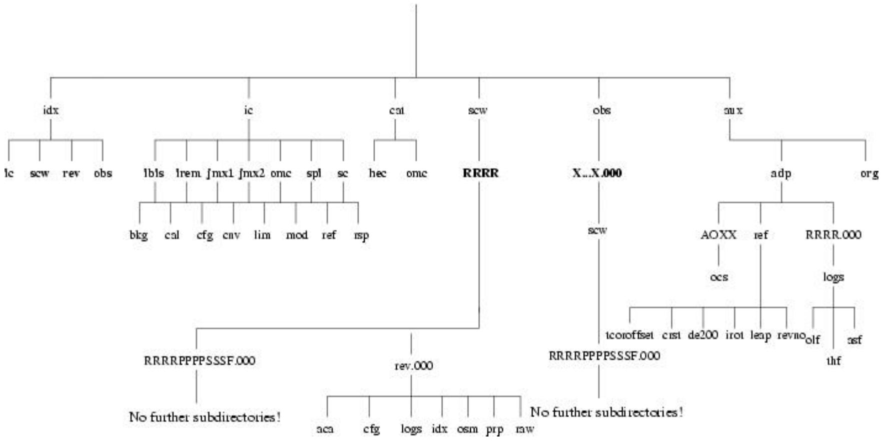

Figure 12: The complete ISDC repository structure of revision 2 and 3 data.

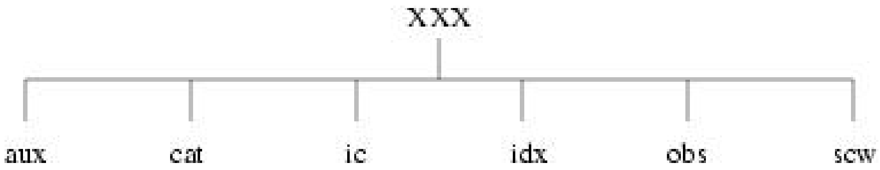

Figure 13: High level directory structure of processed data repository.

by the system pipelines as they process data and for long term storage in the archive. Each processed data repository structure has a high level directory structure as shown in Figure 13.

Where:

- **XXX** represents the name of the processed data repository,
- **aux** contains auxiliary data products received or created by Auxiliary Data Preparation,
- **cat** contains observational catalogs necessary for data analysis,
- **ic** contains data concerning the instrumental calibrations and operational characteristics that are generated offine,
- **idx** contains ISDC index files used for fast searching of data for selected data products,
- **obs** contains results of Science Analysis,
- **scw** contains the results of processing from the science window pipelines on a per-science window basis.

The structure of each processed data repository is defined below.

3.4.2 Auxiliary data: aux

Data files residing under **aux** contain "auxiliary" information that is needed to understand or otherwise make use of the data products derived from the spacecraft telemetry. These data are considered auxiliary because they result from mission planning or on-ground mission analysis, and not from the instrument telemetry. Examples of auxiliary data include the spacecraft attitude and orbit, the time-line, the On-Board Time (OBT) – Universal Time (UTC) correlation, information related to the proposals and the planning, etc.

3.4.3 Catalog data: cat

Data files residing under **cat** contain observational catalogs needed by the ISDC data analysis subsystems and applications, see Section 3.5.

Directory **cat** has two subdirectories. The **hec** subdirectory contains the ISDC high energy source catalogs. The **omc** subdirectory contains OMC source catalogs, as defined and updated by the OMC instrument team, see OMC Analysis User Manual [4] for more details.

3.4.4 Instrument characteristics data: ic

Data files residing under **ic** contain data relating to the calibration and operation of the instruments such as instrument characteristics data (background models, response matrices), housekeeping calibration curves and housekeeping parameter limits. These data files span a wide range of time intervals, are valid for a large number of observation data sets, and are largely instrument specific.

3.4.5 Index file data: idx

Data files stored under **idx** implement various index databases used throughout the ISDC system. These index files are created and manipulated by the INDEX executable set provided by ISDC.

3.4.6 Science Window data: scw

Data files located under the **scw** directory result from ISDC science window and revolution file pipeline processing. Figure 14 shows the directory structure of **scw**.

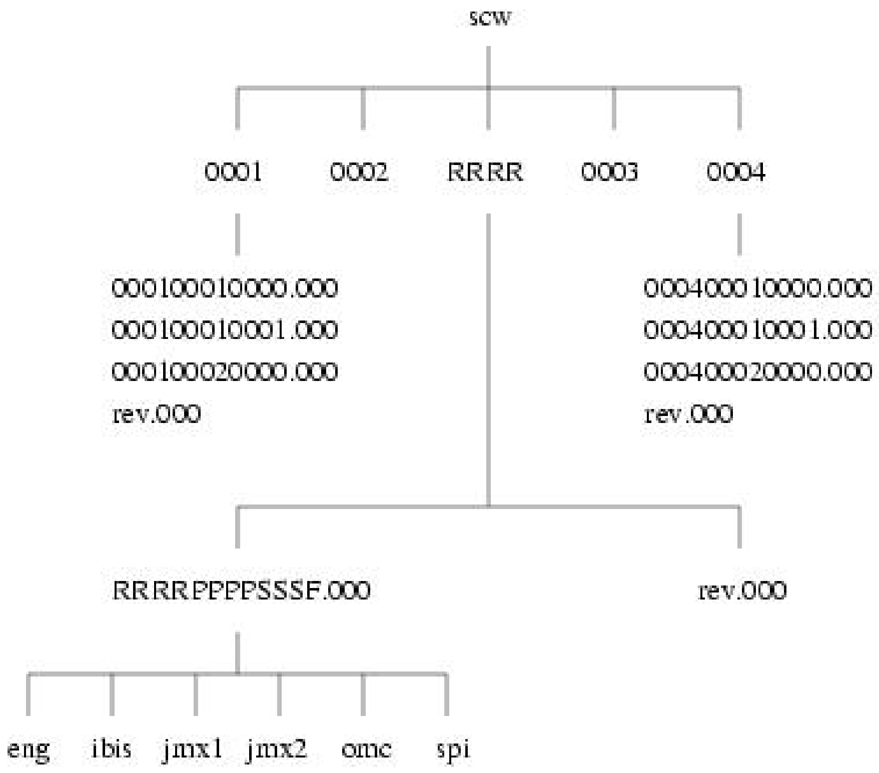

Figure 14: Subdirectory structure of scw for revision 1 data. For revision 2 and 3 data, directory **RRRRPPPPSSSF.VVV** does not have any further subdirectories.

Each of the first-level subdirectories under **scw** has a name of the form **RRRR** corresponding to an orbital revolution number of the S/C as assigned by ISOC, and contains science window data sets derived from telemetry acquired during that revolution. Each **RRRR** directory contains a number of science window and revolution file subdirectories with names of the form **RRRRPPPPSSSF.VVV** and **rev.VVV**, where

- **RRRR** is the revolution number,
- **PPPP** is the ISOC assigned pointing number,
- **SSS** is a sequence used to subdivide pointings or slews during processing if required, and
- **F** is a flag representing the science window type (0 = pointing, 1 = slew, 2 = non-science, values larger than 2 are undefined),
- **VVV** is the version number of the directory, increased with each processing.

The 12 digits RRRRPPPPSSSF form a unique name of a given science window: the Science Window IDentifier (ScWID). The individual science window directories **RRRRPPPPSSSF.VVV** contain data relating to a single science window, whereas the revolution file directories **rev.VVV** contain data derived from a series of science windows from a given revolution.

For revision 1 data, the structure of each **RRRRPPPPSSSF.VVV** consists of a subdirectory for each instrument and one subdirectory eng for engineering data. Under each instrument directory you find the Science Window Group of prepared level, which location you should specify before you start the analysis. The **eng** directory is optional if no engineering-related data exists for the science window.

For revisions 2 and 3, all the data are located directly in **RRRRPPPPSSSF.VVV**, without any further subdirectories.

The **rev.VVV** directories store the revolution file data produced by different subsystems. Under this directory you find data which are generally associated with the instrument configuration and calibrations.

3.4.7 Observation Group data: obs

Data files residing under **obs** result from the data science analysis. Each directory under **obs** contains the products generated for an Observation Group during a single processing run. The directory names are of the form **X. . . X.VVV**, where **X. . . X** is a string of at most 28 characters.

The subdirectory **scw** contains one subdirectory of the form **RRRRPPPPSSSF** (see the explanation of this name in Section 3.4.6) for each science window belonging to the observation group, and consist of all the data products generated for a given science window (i.e., **cor** and **res** data) during Observation Group processing.

3.5 The General Reference Catalog and the Resulting Source List

The General Reference Catalog is the master table listing all known high-energy sources of relevance to *INTEGRAL* . Currently, version 33 of the catalog contains ∼ 2,000 objects. These objects are all those that have been detected by *INTEGRAL* or that are known to be brighter than ∼ 1 mCrab in the 1 keV to 10 MeV band. The catalog is updated regularly when there are enough new sources or changes to warrant a new release. The latest catalog can be retrieved online (http://www.isdc.unige.ch/integral/science/catalogue). At this URL, versions in HTML, TEX, and ASCII are available to complement the FITS file.

The objects in the catalog were compiled from the following sources:

- Macomb & Gehrels gamma-ray sources catalog
- Liu et al. LMXB and HMXB catalogs

- The 4th Uhuru catalog
- Van Paradjis' X-ray Binaries catalog
- HEAO1 A4 catalog
- BATSE observations of Piccinotti's sample of AGN
- Tartarus reduced AGN data
- ASCA Survey catalogs
- Recent IAUCs and ATels
- *INTEGRAL* catalogs (archive, survey, GCDE, etc.)
- The IGR sources page (J. Rodriguez & A. Bodaghee)

In the catalog, you will find information on the name(s) of a source, its position, and error. Identifiers are SIMBAD-compliant and all positions are referenced. Sources in the HTML version have names that link directly to the relevant page in SIMBAD and a position that links to the reference in ADS.

Each source is given a Spectral Energy Distribution (SED) model with parameters such as column density, photon index, and normalization. These models are used to determine counting rates in two bands for ISGRI and JEM-X. This enables the user to select for analysis only sources that are brighter or weaker than a given value. There are a few dozen sources for which models are well understood. We use values from the literature when they are available. In most cases, we assign default models and parameters depending on the source type, and then offer a normalization that can be adjusted to reflect *INTEGRAL* observations that are made public. Table 3 lists the default models and parameters for sources in the catalog. Details of the models are provided after the table. Keep in mind that the models approximate the behavior and do not necessarily reflect actual counting rates.

An instrument-specific flag in the catalog informs the user of sources that have been detected by an *INTEGRAL* instrument. For example, a subset of the catalog can be built by extracting all sources that have been seen by ISGRI. The values of the ISGRI FLAG are:

- **0** Source not detected,
- **1** Source detected,
- **2** Source detected and a position is available with a precision ¡ 3 arcsec.

From version 31 of the catalog a new ISGRI flag (ISGRI_FLAG2) has been introduced, and its values are:

- **0** Source not detected,
- **1** Source detected in a single science window,
- **2** Source detected in a mosaic,
- **5** Very bright source (used for screening of background and images).

Flags for JEM-X, SPI and PICsIT have the value 0 if not detected, and 1 if detected.

Here are two ways to use this feature:

- In FV, you select all rows such that ISGRI FLAG==0 (i.e. not detected by ISGRI), delete the selected rows, and save.
- In OSA, a parameter line similar to the one below will direct the software to concentrate on relevant sources during analysis:

```
CAT_refCat="gnrl_refr_cat_0033.fits[1][ISGRI_FLAG==1]"
```

The examples shown above for the ISGRI FLAG also work for JEMX_FLAG, SPI_FLAG, and PICSIT_FLAG, and they can even be combined:

```
CAT_refCat="gnrl_refr_cat_0033.fits[1][ISGRI_FLAG==1 && SPI_FLAG==1]"
```

The references used to set flags include lists of JEM-X sources, the SPI Bright Source Catalog, survey results, publications, public data from IQLA, and scientists working on private data.

Table 3: Default models and parameters for sources in the *INTEGRAL* General Reference Catalog.

Source Type	SED Model	$N_{\rm H}$	Γ_1	Γ_2	$E_{\rm cut}$	$E_{\rm fold}$	$E_{\rm break}$
		$[10^{22}{\rm cm}^{-2}]$			[keV]	[keV]	[keV]
X-ray Binaries							
LMXB	wabs*cutoff	1.0	1.7	-	5	-	-
LMXB with hardtail	wabs*(cutoff+powerlaw)	1.0	1.7	2.5	10	-	-
HMXB	wabs*highecut*powerlaw	1.0	1.0	-	10	15	-
AGN							
Seyfert I	wabs*cutoff	1.0	1.7	-	100	-	-
Seyfert II	wabs*cutoff	10.0	1.7	-	100	-	-
Blazar	wabs*cutoff	1.0	1.7	-	100	-	-
Quasar	wabs*cutoff	1.0	1.7	-	100	-	-
Radio Pulsars							
Radio Pulsar	wabs*powerlaw	1.0	2.0	-	-	-	-
Unidentified							
X-band detection	wabs*cutoff	1.0	1.7	-	10	-	-
COMPTEL detection	wabs*bknpower	100.0	-1.5	2.1	-	-	1000
EGRET detection	wabs*bknpower	100.0	-1.5	2.1	-	-	10000
INTEGRAL detection	wabs*power (A = 0)	1.0	1.7	-	-	-	-

- **powerlaw** : simple photon power-law, typically due to synchrotron radiation.

$$S(E) = A\left(\frac{E}{E_0}\right)^{\Gamma} \qquad (1)$$

- **cutoff** : power-law with a high-energy exponential cutoff.

$$S(E) = A\left(\frac{E}{E_0}\right)^{\Gamma} \times \exp\left(\frac{-E}{E_{\rm cut}}\right) \qquad (2)$$

- **bknpower** : broken power-law.

$$S(E < E_{\rm break}) = A\left(\frac{E}{E_0}\right)^{\Gamma_1} \qquad (3)$$

$$S(E \geq E_{\rm break}) = A\,E_{\rm break}^{(\Gamma_2-\Gamma_1)} \times \left(\frac{E}{E_0}\right)^{\Gamma_2} \qquad (4)$$

- **wabs** : photoelectric absorption using Wisconsin cross-sections.

$$m(E) = \exp\left(-N_H\sigma(E)\right) \qquad (5)$$

- **highecut** : high-energy cutoff.

$$m(E < E_{\rm cut}) = 1 \qquad (6)$$

$$m(E \geq E_{\rm cut}) = \exp\left(\frac{E_{\rm cut} - E}{E_{\rm fold}}\right) \qquad (7)$$

E	energy [keV]
E_0	reference energy [keV] (1 keV in XSPEC)
E_{cut}	exponential cutoff energy [keV]
E_{fold}	exponential folding energy [keV]
E_{break}	break energy [keV]
Γ_1	photon index for$E < E_{\mathrm{break}}$
Γ_2	photon index for $E \geq E_{\mathrm{break}}$
N_{H}	column density of hydrogen [10^{22}atoms cm^{-2}]
$S(E)$	monochromatic flux at energy E [photons $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$]
$A = S(E_0)$	normalization ($F_{\mathrm{obs}}/F_{\mathrm{comp}}$) at 1 keV [photons $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}\,\mathrm{keV}^{-1}$]
$m(E)$	absorption coefficient of $S(E)$ at energy E
$\sigma(E)$	photoelectric cross section (excluding Thompson scattering)

In Table 4, the content of the catalog is given. Columns in bold are the minimum you should fill to add a new source into the catalog. For the **SOURCE_ID**, you can enter whatever you want, but make sure that this entry is unique. The class codes for the General Reference Catalog can be found online (`http://heasarc.gsfc.nasa.gov/W3Browse/catalog/class.html`). Note that the FITS format of the catalog truncates names and comments that are longer than 20 and 128 characters, respectively. For this reason, sources such as 1RXS or XMMU will lack the last digit in their name. Users are advised to refer to the HTML version for complete names and comments (`http://www.isdc.unige.ch/integral/science/catalogue`).

During the analysis, the Catalog Source List, which contains the list of sources observed by an *INTEGRAL* instrument in the given field of view, is extracted from the General Reference Catalog with the `cat_extract` program. The resulting Catalog Source List is in a similar format as the General Reference Catalog, but, starting at this level, the distinction between the different instruments is made. In Table 5 you will find descriptions of the columns that are added to the output catalogs.

3.6 The Scientific Analysis and Scientific Analysis Levels

The Scientific Analysis denotes the processing required to turn the prepared data into the final high-level data products, such as sky images, source positions, fluxes, spectra and light curves. The scientific analysis is designed to produce data that can be used as inputs into external packages.

The scientific analysis pipelines are made of independent components for each of the *INTEGRAL* instruments. It is possible to independently execute just one of these components, i.e., to analyze the data from one instrument only.

Taking advantage of the scientific analysis similarities, tasks/levels common to all instruments are identified, and designated with a unique name. This makes a system more user friendly, avoiding confusion between different instruments. A given name refers to similar tasks and conversely. The only exception is IMA2 level, which combines different tasks for OMC, JEM-X and IBIS.

The definition of common scientific analysis levels also enforces similarities in the Architectural Design of the instrument specific scientific analysis. However, the scientific analysis is partly instrument specific. Some tasks/levels do not exist for all instruments, and the task order also differs. These different levels are described below.

Table 4: Content of the **GNRL-REFR-CAT** Data Structure.

Column Name	Description
SOURCE_ID	ISDC unique source identifier
DAY_ID	Modified Julian Date of source's first identification
NAME	One commonly used name for the source
CLASS	Source classification code
RA_OBJ	Source right ascension in degrees
DEC_OBJ	Source declination in degrees
ERR_RAD	Error radius in degrees
ISGRI_FLAG	ISGRI detection flag (0:undetected, 1:detected, 2:source detected and a position is available with a precision < 3")
ISGRI_FLAG2	ISGRI detection flag (0:undetected,1:detected in a single science window,2:detected in a mosaic,5:very bright source, used for screening of background and images)
JEMX_FLAG	JEMX detection flag (0:undetected, 1:detected)
SPI_FLAG	SPI detection flag (0:undetected, 1:detected)
PICSIT_FLAG	PICsIT detection flag (0:undetected, 1:detected)
	data for simulator
SPA_MODL	Model for source spatial extension (point, disk, ellipse, square, gaussian, Bspline, etc.)
SPA_NPAR	Number of parameters for source spatial extension
SPA_PARS	Parameters for source spatial extension
SPE_MODL	Model for source spectrum (XSPEC syntax)
SPE_NPAR	Number of parameters for source spectrum
SPE_PARS	Parameters for source spectrum
VAR_MODL	Model for source intensity variability (const, sin, burst)
VAR_NPAR	Number of parameters for source intensity variability
VAR_PARS	Parameters for source intensity variability
COMMENTS	Aliases, comments
	selection parameters which can be used by `cat_extract`
SPI_FLUX_1	SPI flux (photon/s/cm2) in the soft SPI energy band (15 – 300 keV)
SPI_FLUX_2	SPI flux (photon/s/cm2) in the hard SPI energy band (300 – 8000 keV)
ISGR_FLUX_1	ISGRI flux (count/s) in the soft ISGRI energy band (20 – 60 keV)
ISGR_FLUX_2	ISGRI flux (count/s) in the hard ISGRI energy band (60 – 200 keV)
PICS_FLUX_1	PICsIT flux (count/s) in the soft PICsIT energy band (100 – 600 keV)
PICS_FLUX_2	PICsIT flux (count/s) in the hard PICsIT energy band (600 – 6000 keV)
JEMX_FLUX_1	JEM-X flux (count/s) in the soft JEM-X energy band (3 – 10 keV)
JEMX_FLUX_2	JEM-X flux (count/s) in the hard JEM-X energy band (10 – 50 keV)
E_MIN	Vector of lower energy boundaries (keV)
E_MAX	Vector of upper energy boundaries
FLUX	Vector of flux values (photon/s/cm2)
FLUX_ERR	Vector of flux errors (photon/s/cm2)
SEL_FLAG	Source selection flag - not filled at the moment
FLAG	Generic flag

COR – Data Correction IBIS, SPI, JEM-X, OMC

corrects science data for instrumental effects, such as energy and position corrections.

POIN – Pointing Definition SPI

Table 5: Columns added to the **XXXX-SRCL-RES** Data Structure.

Column Name	Description
	Columns added in the output catalogs of all instruments
DETSIG	Source detection significance
RELDIST	Relative distance of the source (= d/(r1 + r2), where d is the distance to the catalog source identified with the detected source and r1, r2 are the respective error radii of the sources)
	Columns added to the output ISGRI catalog (**ISGR-SRCL-RES**)
NEW_SOURCE	New source flag (0 if old, 1 if new)
RA_FIN	Right Ascension of the found position of the source
DEC_FIN	Declination of the found position of the source
FIN_RD_ERR	Error of the found position of the source in RA and DEC
SCW_NUM_C	number of times that the source was in ScW FOV. If SCW_NUM_C = 0 then it is a new source
SCW_NUM F	number of times that the source was found in ScW image
OG_NUM	number of times that the source was found in OG image
	Columns added to the output JEM-X catalog (**JEMX-SRCL-RES**)
COSX_JMX	Direction cosine between X-axis and source in instrument coordinates
COSY_JMX	Direction cosine between Y-axis and source in instrument coordinates
COSZ_JMX	Direction cosine between Z-axis and source in instrument coordinates

defines a number of time periods during which the S/C attitude can be considered as constant for SPI analysis (e.g., a slew can be split into a number of periods) and extracts the spacecraft attitude information from the input group. Stores the results in a pointing file for further use.

GTI – Good Time Handling IBIS, SPI, JEM-X, OMC

generates, selects, and merges Good Time Intervals to produce a unique GTI which is to be used for selecting good events. GTI can be used to exclude times of TM stream gaps, high background, instrument anomalous behavior, etc. The net observing time is also computed in this step.

DEAD – Dead and Live Times IBIS, SPI, JEM-X

computes dead time (i.e., the time during which the instrument was not capable to register the incoming photons for different reasons) within the good time interval.

BIN_I – Event Binning for Imaging IBIS, SPI, JEM-X

defines the energy bins to be used for imaging, selects good events within the GTI, and creates event arrays.

BKG, BKG_I, BKG_S – Background Model Generation IBIS, SPI, JEM-X

derives estimated background from models, either for imaging analysis (BKG I), for spectral analysis (BKG S), or for the general case (BKG). Corrects IBIS shadowgrams for background.

CAT, CAT_I, CAT_S – Catalog Source Selection IBIS, SPI, JEM-X selects a list of known sources from the ISDC reference catalog, or from previous analyses, and creates a source data structure, containing source location and expected flux values.

IMA – Image Reconstruction IBIS, SPI, JEM-X, OMC

generates sky images and searches for significant sources. If sources are detected, a new source data structure is created, including a copy of the input source data and the newly derived source properties, such as source positions and fluxes.

IMA2 – PICsIT Image Reconstruction IBIS

For IBIS, ISGRI and PICsIT image analysis are split into two levels.

IMA2 – JEM-X Mosaic creation JEM-X

At this level the mosaic image of JEM-X observations along with the summary list of all found sources are created.

IMA2 – OMC Summary List OMC

At this level the summary list of the derived magnitudes for all OMC shots is created.

BIN_S – Event Binning for Spectral Analysis IBIS, SPI, JEM-X

defines the energy bins to be used for spectral analysis, selects good events within the GTI, and creates detector spectra.

SPE – Spectra Extraction IBIS, JEM-X

extracts spectra for individual sources, and produces the specific response files (ARFs) needed for spectral fitting with the XSPEC package.

BIN_T – Create Detector Light Curves JEM-X

defines the energy bins to be used for light curves analysis,

LCR – Extract Source Light Curves IBIS,JEM-X

produces light curves for individual sources.

3.7 ISDC Julian Date

The Julian Date is a standard way of expressing Earth time. The definition of the Julian Date depends however on the time system, i.e., there are different possible Julian Dates. ISDC has chosen to use the Terrestrial Time. The Julian Date is defined as the number of days elapsed since December 31 of year -4713 at noon. Unfortunately, the present Julian Date is about 2.5 million days. Expressing times with millisecond accuracy would require 17 significant digits, which does not fit in a double precision real. The Modified Julian Date (MJD) is equivalent to the Julian Date -2 400 000.5 days. Current MJD is about 55 000, and two digits can be spared. This would marginally fit in a double precision real.

To avoid any accuracy problem caused by the storage of the data in a double precision real, the ISDC Julian Date (IJD) was defined, which is the Julian Date, but with a reference close to the date of the *INTEGRAL* mission. The 1st January of Year 2000 at 0h 0m 0s Terrestrial Time has been chosen as the reference Julian Date.

Since Terrestrial Time differs from UTC by 32.183 sec + 32 leap seconds at the start of year 2000, the UTC origin of the IJD is actually 1999-12-31T23:58:55.817. IJD is connected to the MJD in the following way: IJD = MJD - 51544. The tool `converttime` distributed with OSA performs the time conversions from different systems.

4 Software Tools

4.1 Standard Tools

Here we mention the packages that can be useful to analyze the results of the *INTEGRAL* scientific analysis.

XSPEC

XSPEC is an X-ray spectral fitting package. The XSPEC User's Guide can be found at

`http://heasarc.gsfc.nasa.gov/docs/xanadu/xspec/index.html`

At the same place you will find a link to the "download and install" instructions page.

FTOOLS

FTOOLS is a collection of utility programs used to create, examine, or modify the contents of FITS data files. The FTOOLS User's Guide can be found at

`http://heasarc.gsfc.nasa.gov/ftools/`

The package itself can be also downloaded from this page.

FV

Among different components of FTOOLS we want to draw your attention to the program FV. FV is a software tool for viewing and editing any FITS format image or table. You can learn more about it and download at

`http://heasarc.gsfc.nasa.gov/docs/software/ftools/fv/`

DS9

DS9 is an astronomical imaging and data visualization application. DS9 supports FITS images and binary tables, multiple frame buffers, region manipulation, and many scale algorithms and colormaps.

You can download it and find more information at

`http://hea-www.harvard.edu/RD/ds9/`

ISDCroot

ISDCroot is the CERN ROOT system (`http://root.cern.ch`) with ISDC additions. The first addition is that all of the ISDC support libraries are available in ROOT so that scripts which use DAL or DAL3 library calls can be written. The second addition is that *INTEGRAL* analysis executables and/or any FTOOLS-like executable can be easily run from within ISDCroot. The final addition is IC access routines which allow easy access to the IC tree. With the above combinations, it is easy to write scripts to read data from ISDC data repositories into ROOT analysis classes. Once this is done, one can easily display and manipulate data for analysis.

4.2 Scripts and Executables

As we already discussed (section 3.6) the *INTEGRAL* software for scientific analysis is composed of small blocks combined into bigger ones. These small blocks are called executables. The executables are combined into intermediate scripts which, in their turn, form the content of a main script, one per instrument.

The simplest way to run Scientific Analysis for a specific instrument is to set the parameters you are interested in, and to run the corresponding main script. There are several ways of entering the parameters: you can either input them from the command line, or use a Graphical User Interface (GUI), which is launched automatically when you run the script. From the command line you should enter those parameters that are not defined as hidden in the parameter file (see below), and those hidden parameters that you want to change. With a GUI you can see all parameters selecting the option "Show hidden parameters". The place from where the parameter files are taken is defined by the value of the PFILES environment. It has the following structure:

```
echo $PFILES
local/directories/pfiles;system/directories/pfiles:system/directories/pfiles
```

First the parameter file is looked for in the local directories (from left to right), and then, if not found, in the system directories. In the later case the parameter file is copied to the local directory. If you have changed the OSA version on your computer, you should delete all parameter files from your local pfiles directory to avoid inconsistency of the parameters.

4.2.1 Parameter Files

The parameter file is an ASCII text file where each line contains a parameter description with the following simple format:

Parameter,Type,Query,Value,From,Until,Prompt

where

- **Parameter** is the parameter name such as `input-file`.
- **Type** is a single character, describing whether the parameter is a text string (`s`), boolean expression (`b`), real (`r`) or integer (`i`) number.
- **Query** is a flag `q` or `h` to prompt the user for input of the parameter value.
  `q` will display the prompt text in `Prompt` and wait for manual input.
  `h` means the parameter is hidden from external input with a value given by `Value`.
  If after the flag there is letter `l`, i.e., `ql` or `hl` then the program remembers the last input and suggests it to you during next run.
- **Value** is the actual value of the parameter depending on its type `Type`.
  `s` expects a text string in quotes such as "filename".
  `r` expects a real decimal such as 12.345.
  `i` expects an integer number such as 123.
  `b` expects either `yes` or `no`.
- **From** is a lower limit on the parameter value in real or integer format.
- **Until** is an upper limit on the parameter value in real or integer format.
- **Prompt** is a display prompt text used to ask for manual input if `Query` is `q` and should be a text string in quotes such as "Enter name of input observation file: ".

4.2.2 Parameter Types

The parameters required by the script governing the Scientific Analyses can be divided into four parts.

1. **Observation Group**

   The first thing you will be asked by the script is the DOL of the Observation Group you want to work with. During the creation of this group, you have specified the list of the science windows you want to work with and they were attached to the created Observation Group. During script runs, the newly created files will be attached to the Observation Group.

2. **IC Master Group**

   To analyze your scientific data you also need to specify the location of the files with Instrument Characteristics (IC), such as the instruments calibration, instrument responses, instrument background etc. These files can exist in different versions – for example if you are trying to apply different algorithms and to check which one is better. The version number is reflected by the value of the VERSION keyword in the header unit of the corresponding FITS file. If the chosen algorithm makes use of the parameters varying with time then with its help several files will be produced, each valid for a given period of time indicated with the STRT_VAL and END_VAL keywords. The numbers in the name of the FITS file reflects neither the version number, nor the time period but only the order they were added to the database.

   In order to avoid entering the location of the tons of IC files by hand the IC Master Group was created. One of the members of this group is a version table specifying which version of each IC file should be used in analysis. In principle, it is possible that for different purposes different sets of versions are preferable. In this case, this table will have several rows with different MNEMONIC names and for this reason such a name should be specified in the script along with the location of the IC Master Group. If these parameters are specified, then the IC master Group returns to the script the location of the IC files with a corresponding version and the time period corresponding to the science window under the analysis.

3. **Calibration data**

   Besides the location of Science Data and IC files, location of the calibration data, provided by the system of on-board calibration sources, should also be specified. The time history of these data is not connected with the science windows and thus, these data are not attached to the Science Window Group.

4. **Reference Catalog**

   The next thing that you should specify is the DOL of the reference catalog. This catalog will be used by the analysis software as a starting point which increases the speed of the source location. As a result of the analysis, a new catalog based on the analyzed sky region will be produced.

5. **Parameters not connected with the file locations**

   And of course in the script parameter file you should specify such parameters as: aim of analysis, desired energy ranges, background fitting method, etc. The detailed description of the parameters are given in the instrument specific user manuals [1, 2, 3, 4], see chapters Scientific Analysis and Cookbook.

4.3 CFITSIO Table Row Selection

In many of ISDC tools, you have a possibility to apply a selection string to choose only data relevant for your task. Row selection allows applications to build an algebraic expression that evaluates to TRUE or FALSE for each row of a table element, or set of table elements. If the expression is TRUE for a given row, then that row is selected (the result of selection is API function specific), otherwise it is not selected. The expression can be an arbitrarily complex series of operations performed on constants, attribute values, and table columns for a given table element. Below we describe the details of the row selection syntax, this part was mostly taken from the Data Access Layer Users Guide, which you may find in `$ISDC_ENV/help/dal.ps`.

4.3.1 Basic Table Row Selection

Attributes and columns are referenced by name in the expression. Any string of characters not surrounded by quotes (i.e., a constant string) or followed by open parentheses (i.e., a function name) will be initially interpreted as a table column name and its content for the current row will be inserted into the expression before evaluation. If no such table column exists, an attribute of that name will be searched for – and its value used, if found. To force the string to be interpreted as an attribute name (in case there is both a table column and attribute with the same name), precede the attribute name with a single pound sign, '#', as in '#RA'. If the column or attribute name contains a space or a character which might appear as an arithmetic term then enclose the name in '$' characters as in $MAX-PHA$ or #$MAX-PHA$. Names are case insensitive. The following boolean operators are available:

"equal"	`.eq. .EQ. ==`	"not equal"	`.ne. .NE. !=`
"less than"	`.lt. .LT.` <	"less than/equal"	`.le. .LE.` <= =<
"greater than"	`.gt. .GT.` >	"greater than/equal"	`.ge. .GE.` >= =>
"or"	`.or. .OR.` ||	"and"	`.and. .AND.` &&
"negation"	`.not. .NOT. !`	"approx. equal(1e-7)"	~

The expression may also include arithmetic operators and functions. Trigonometric functions use radians, not degrees. The following arithmetic operators and functions can be used in the expression (function names are case insensitive):

"addition"	`+`	"subtraction"	`-`
"multiplication"	*	"division"	`/`
"negation"	`-`	"exponentiation"	** ^
"absolute value"	`abs(x)`	"cosine"	`cos(x)`
"sine"	`sin(x)`	"tangent"	`tan(x)`
"arc cosine"	`arccos(x)`	"arc sine"	`arcsin(x)`
"arc tangent"	`arctan(x)`	"arc tangent"	`arctan2(x,y)`
"exponential"	`exp(x)`	"square root"	`sqrt(x)`
"natural log"	`log(x)`	"common log"	`log10(x)`
"modulus"	i % j	"random # [0.0,1.0)"	`random()`

Conditional arithmetic can be performed by multiplying, '*', boolean and arithmetic expressions together. If the boolean subexpression evaluates to TRUE, the larger expression has the value of the arithmetic subexpression. If the boolean is FALSE, the expression evaluates to zero. For example, $7 \times (5 > 3)$ equals 7 whereas $7 \times (5 < 3)$ equals 0. In addition, several constants are built in for use in numerical expressions:

`#pi`	`3.1415...`	`#e`	`2.7182...`
`#deg`	`#pi/180`	`#row`	current row number

A string constant must be enclosed in quotes as in 'Crab'.

Table columns with vector bins (i.e., more than one data value per bin) can also be used in building the expression. No special syntax is required to operate on all data values of the bin. Simply use the table column name as for a scalar column. Vector bin columns can be freely intermixed with scalar bin columns or constants in virtually all expressions. The result will be of the same dimension as the vector. Two vector bin columns in an expression, though, need to have the same number of data values.

Arithmetic and logical operations are all performed on a data value by data value basis. Two special functions are available to operate on table columns with vector bins: `SUM(x)` and `NELEM(x)`. The former literally sums all the data values belonging to the bin of column X, returning a scalar value. If X is column of DAL data type DAL_BOOL, `SUM(x)` returns the number of TRUE elements. The latter, `NELEM()`, returns the number of data values in each bin of column X (`NELEM()` also operates on DAL_BIT and DAL_CHAR columns, returning their column widths). As an example, to test whether all data values of a bin vector satisfy a given logical comparison, one can use the expression:

```
SUM( COL1 > COL2 ) == NELEM( COL1 )
```

which will return TRUE if all data values in the bin of table column COL1 are greater than their corresponding data values in COL2.

Note that table columns with variable length bins are not supported under the algebraic expression syntax.

Some examples of basic row filter expressions follow. All string quantities represent table column names or attribute names in the table elements to be evaluated. Remember that each of these expressions is to evaluate to either TRUE or FALSE for a given table row.

`[TIME > 123456 && TIME <= 555555]`	- Extract all table rows where the column TIME has values between 123456 and 555556 (exclusive)
`[#row >= 125 && #row <= 175]`	- Extract row numbers 125 through 175 from all tables examined
`[OBJECT  == 'CRAB']`	- Extract all table rows if the attribute "OBJECT" has the value of "CRAB"
`[abs(sin(theta * #deg)) < 0.5 ]`	- Extract all table rows having the absolute value of the sine of THETA less than 0.5 where the angles are tabulated in degrees
`[SUM( SPEC > 3*BACKGRND) >=1]`	- Extract all table rows containing a spectrum, held in vector column SPEC, with at least one data value 3 times greater than the background level held in the attribute BACKGRND
`[log(PHA) - log(PHI) > 3.75]`	- Extract all table rows where the difference between the log of the PHA and PHI column values is greater than 3.75

`[X ∼ 36.3 && Y ∼ 05.34 &&`
`sin(#ROTATION) < pi/2.0]`
	- Extract all table rows with detector X coordinates of approx. 36.3, Y detector coordinates of approx. 5.34 and the sine of the image rotation is less than 90 degrees

4.3.2 Advanced Table Row Selection

There are three functions that may be used in the algebraic expression that are borrowed from the SAO region filtering syntax. They return a boolean true or false depending on whether a two dimensional point is in the region or not:

“point in a circular region”

```
circle(xcntr,ycntr,radius,Xcolumn,Ycolumn)
```

“point in an elliptical region”

```
ellipse(xcntr,ycntr,xhlf_wdth,yhlf_wdth,rotation,Xcolumn,Ycolumn)
```

“point in a rectangular region”

```
box(xcntr,ycntr,xfll_wdth,yfll_wdth,rotation,Xcolumn,Ycolumn)
```

where

`(xcntr,ycntr)` are the (x,y) position of the center of the region
`(xhlf_wdth,yhlf_wdth)` are the (x,y) half widths of the region
`(xfll_wdth,yfll_wdth)` are the (x,y) full widths of the region
`(radius)` is half the diameter of the circle
`(rotation)` is the angle(degrees) that the region is rotated with respect to (xcntr,ycntr)
`(Xcoord,Ycoord)` are the (x,y) coordinates to test, usually column names
NOTE: each parameter can itself be an expression, not merely a column name or constant.

There is also a function for testing if two data values are close to each other, i.e., if they are “near” each other to within a user specified tolerance. The arguments value 1 and value 2 can be integer or real, and represent the two data values whose proximity is being tested to be within the specified tolerance – also integer or real:

```
near(value_1, value_2, tolerance)
```

When a NULL, or undefined, value is encountered in the table element, the expression will evaluate to NULL unless the undefined value is not actually required for evaluation, e.g., “TRUE .or. NULL” evaluates to TRUE. The following two functions allow some NULL detection and handling: `ISNULL(x)` and `DEFNULL(x,y)`. The former returns a boolean value of TRUE if the argument x is NULL. The latter defines a value to be substituted for NULL values; it returns the value of x, if x is not NULL, otherwise it returns the value of y.

The following type casting operators are available, where the enclosing parentheses are required and taken from the C language usage. Also, the integer to real casts values to double precision:

”real to integer”	`(int) x`	`(INT) x`
”integer to real”	`(float) i`	`(FLOAT) i`

Bit masks can be used to select out rows from table columns of DAL data type DAL_BIT. To represent the mask, binary, octal, and hex formats are allowed:

binary:	`b0110xx1010000101xxxx0001`	
octal:	`o720x1 ->`	`(b111010000xxx001)`
hex:	`h0FxD ->`	`(b00001111xxxx1101)`

In all the representations, an x or X is allowed in the mask as a wild card. Note that the x represents a different number of wild card bits in each representation. All representations are case insensitive.

To construct the boolean expression using the mask as the boolean equal operator described above on a bit table column. For example, if you had a 7 bit table column with name “FLAGS” and wanted all rows having the bit pattern 0010011, the selection expression would be:

```
flags == b0010011
```

or

```
flags .eq. b10011
```

It is also possible to test if a range of bits is less than, less than equal, greater than and greater than equal to a particular boolean value:

```
flags  <=  bxxx010xx
flags .gt. bxxx100xx
flags .le. b1xxxxxxx
```

Notice the use of the x bit value to limit the range of bits being compared. It is not necessary to specify the leading (most significant) zero (0) bits in the mask, as shown in the second expression above. Bit wise AND, OR and NOT operations are also possible on two or more bit fields using the '`&`' (AND), '`|`' (OR), and '`!`' (NOT) operators. All of these operators result in a bit field which can then be used with the "equal" operator. For example:

```
(!flags) == b1101100
(flags & b1000001) == bx000001
```

Bit fields can be appended as well using the '`+`'operator. Strings can be concatenated this way, too.

4.4 Toolbox

Here we give the brief description of the generic OSA tools. More information could be seen in the .txt file or by entering the name of the program with the `--h` option after it, e.g.

```
dal_attach --h
```

Table 6: List of generic tools (more often used part).

Name	Description
barycent	Converts time stamps of events or light curves to barycentered times assuming that the data comes from a given source direction.
cat2ds9	Creates DS9 region file from the list of sources. This can be a catalog extract, or a list of sources produced by the analysis. See more details in section 6.3.
cat_merge	Merges two source lists, e.g., two extracts from different catalogs.
converttime	Reads input time in a user specified format and converts it to either user specified output format or into all formats known by the converttime.
dal_create	Creates a FITS file according to a template.
evts_extract	Creates an instrument independent event list.
evts_pick	Combines event data distributed over several Science Windows and tables within the Science Window into a single table. Limited to ISGRI and JEM-X in this release.
gti_attitude	Creates a GTI depending on attitude stability, see Science Analysis sections of

	[1], [2],[3],[4] for more details on practical implementations.
gti_create	Generates GTIs depending on housekeeping parameters, see Science Analysis sections of [1], [2],[3],[4] for more details on practical implementations.
gti_data_gaps	Generates GTIs depending on the not missing science data packets, see Science Analysis sections of [1], [2],[3],[4] for more details on practical implementations.
gti_import	Imports User GTIs into the ISDC GTI index files, see Science Analysis sections of [1], [2],[3],[4] for more details on practical implementations.
gti_merge	Merges Good Time Intervals,see Science Analysis sections of [1], [2],[3],[4] for more details on practical implementations.
gti_pick	Creates a user GTI file from ISDC GTIs.
gti_user	Creates user GTI table, see 6.2 for more details.
image_mosaic	Combines many count rate sky images into a mosaic count rate image and an exposure map.
lc_pick	Selects lightcurves created for a given source and combines them in a single file. Either as individual tables combined by an Index (the default) or merged together to a single dataset. See Cookbooks of [1] and[3] for examples on practical usage.
og_create	Builds the OG from an index of prepared science window groups and observation IDs, setting up the complete directory structure; see 6.1 for more details.
og_clean	This executable will clean an OG up to a certain level. To do that, it will erase the files from the disk. To work properly, the group should not have been used to build another OG or groups of higher level.
og_copy	Copies the OG.
q_identify_srcs	Compares sources found by the software with the catalog.
spe_pick	Combines the extracted spectra for a given source across multiple Science Windows. See Cookbooks of [1] and [3] for examples on practical usage.
src_collect	Combines source data obtained by the imaging analysis for individual Science Windows into a single table covering an Observation Group. It is possible to retrieve only selected results, e.g., just for a given source.
txt2idx	All DOLs contained in the ASCII list will be added to the index. See Cookbooks of [1] and [3] for examples on practical usage.

Table 7: List of generic tools (less often used part).

Name	Description
dal_append	Appends a data structure to a group.

dal_attach	Attaches up to 5 children elements to one DAL group.
dal_attr	Reads, writes or deletes attributes.
dal_attr_copy	Copies an attribute from a DOL to a DOL.
dal_clean	Clean out group tables and children.
dal_copy	Reorganizes group, localizing the given subgroup to a separate FITS file.
dal_detach	Detaches and, upon request, deletes data structures from a group. Data structures are only detached (and deleted) when they do not contain any data structures from the input object anymore. Data structures that should be deleted are deleted only if they are not claimed by another object.
dal_dump	Dumps one column of a FITS table to STDOUT or a log file.
dal_list	Lists all the children of DAL groups.
dal_grp_extract	Creates a new group in a separate FITS file, detaching all the children from the old group.
dal_sort	Sorts a table using one or more columns.
dal_verify	Performs simple verification of DAL groups.
ic2dol	Gets the DOLs of the selected element of the INDEX.
ic_find	Selects objects and sorts the rows in an INDEX of IC data.
idx_add	Adds an entry to the index.
idx_collect	Creates a subindex from an existing index using a selection expression based on the columns of the index. It is also possible to create a subindex of children of the input index which may be of a different data structure; the executable will decend into the child objects to find the correct structure specified by the output file template.
idx_find	Creates an index of the primary index elements conform to the given criteria. The resulting index can be sorted on a set of columns.
idx_merge	Merges two indices.
isdc_dircmp	Examines **dir1** and **dir2** and generates various tabulated information about the contents of the directories.

4.5 Known limitations of the generic tools

- (revision 1 and 2 of the archive) In addition to the time correlation corrections listed in Walter et al, 2003 (A&A 411, L25) a new correction should be added on data taken between 20th August 2003 (rev. 104) and 29th June 2004 (rev. 209) . All UTC (IJD) time stamps derived from on board time should be increased by 385 microseconds. This correction is taken into account in reprocessed data (revision 2 of the archive).

- In revision 3 of the archive, these corrections are not necessary.

- *lc_pick*: since template version 7.7 a new column was added to the JMX[1,2]-SRC.-LCR and JMX[1,2]-DETE-LCR templates. Therefore the new *lc_pick* (version 3 and above) cannot work with old data (produced with template version < 7.7, like OSA 4.2) as input data.

- Make sure that the FTOOLS Package (`http://ftools.gsfc.nasa.gov`) release you use is 5.3 or newer.

- Spectral analysis tools which are strict with OGIP definitions, e.g., SHERPA or ISIS, refuse to read the OSA produced spectra, because they use a non-standard naming of the extensions and of some keywords. `Xspec` has been hacked by its developers to overcome this issue.

5 Setting up the environment

This section describes how to set up the environment to successfully run the ISDC software.

The following environment variables need to be set by the user before running any ISDC software:

1. **ROOTSYS**

   This environment variable has to point to the location where the ROOT software package used for OSA is installed.

   In general, the ROOTSYS directory will at least contain the following subdirectories:

   - bin
   - cint
   - include
   - lib
   - man

   If you are not sure about your local ROOT installation, contact your system administrator. The ROOT software package can be downloaded from the ISDC WWW pages at the following URL: `http://www.isdc.unige.ch/integral/analysis#Software`

2. **ISDC_ENV**

   This environment variable has to point to the location where the ISDC Off-line Scientific Analysis Software is installed.

   In general, the ISDC_ENV directory will at least contain the following subdirectories:

   - bin
   - help
   - f90mod
   - include
   - lib
   - pfiles
   - share
   - templates

   The ISDC Off-line Scientific Analysis software package can be downloaded from the ISDC WWW pages at the following URL: `http://www.isdc.unige.ch/integral/analysis#Software`

3. **REP_BASE_PROD**

   This environment variable has to point to an ISDC compliant data repository, i.e. the location where the ISDC Data and Instrument Characteristics packages are installed.

   In general the REP_BASE_PROD directory will at least contain the following subdirectories (or symbolic links):

   - aux - contains 'Auxiliary Data'
   - cat - contains 'High Energy' and 'OMC' Catalogs
   - ic - contains 'Instrument Characteristcis'
   - idx - contains 'Indexes'
   - scw - contains ISDC data corresponding to individual science windows up to the PRP level
   - obs - contains science level observation data

The ISDC Instrument Characteristics package can be downloaded from the ISDC WWW pages at the following URL: http://www.isdc.unige.ch/integral/analysis#Software

4. **ISDC_REF_CAT**

   This environment variable has to contain the 'Data Object Locator' (DOL) of the ISDC Catalog. For the description of a DOL see the section on '*INTEGRAL* Data' of this document.

   An example could be $REP_BASE_PROD/cat/hec/gnrl_refr_cat_0033.fits

   Note: You should type `echo "$ISDC_REF_CAT"` to verify the setting of this environment variable. The double quotes (") are significant.

   The ISDC Catalog can be obtained from the following URL:
   http://www.isdc.unige.ch/integral/analysis#Software

Execute

```
setenv my_variable "value"
```

to set an environment variable with the name my_variable to the value my_value. The double quotes (") may be significant in some cases, so it is always good to use double quotes.
If the `setenv` command fails with a message like: `'setenv: command not found'` or `'setenv: not found'`, you are very likely not using a command interpreter (shell) of a C-Shell flavour (e.g., csh or tcsh). Please replace the single command

```
setenv my_variable my_value
```

by the following command sequence

```
my_variable=my_value ; export my_variable
```

Warning: No blank characters must be present in the `my_variable=my_value` command. Otherwise you will get an error message like `my_variable not found`.

When these four environment variables are set, the following script should be executed to complete the set-up:

```
source $ISDC_ENV/bin/isdc_init_env.csh
```

You should see no errors or warning messages from this script.

To learn more about this script in general, its warning and error messages and the error recovery, type:

```
source $ISDC_ENV/bin/isdc_init_env.csh help
```

After the successful completion of the isdc_init_env script the environment is ready to run the ISDC software. Please see the cookbook sections of the instrument specific Analysis User Manual for details on how to run the ISDC Off-line Scientific Analysis Software.

If the `source` command fails with a message like: `'source: not found'` or you get many error messages indication that the command actually failed, you are very likely not using a command interpreter (shell) of a C-Shell flavour (e.g., csh or tcsh). Please replace the single command

```
source $ISDC ENV/bin/isdc_init_env.csh
```

by the following single command

```
. $ISDC ENV/bin/isdc_init_env.sh
```

Warning: You really must type the '`.`' followed by a blank character before the name of the script. Otherwise the setting of the environment variables inside the script will not be kept after the script terminated.
In addition, make sure to replace the script extension `.csh` by `.sh`

6 Data Analysis

In this section we give a brief overview of the *INTEGRAL* data analysis methodology. The analysis is achieved by four independent instrument specific analysis chains (scripts). All of them follow the same approach. First thing to do is to select the science windows you want to analyze. From the list of the selected Science Windows you create the Observation Group (Table 8). If you want to analyze only some time periods within your Observation group you define your own good time intervals (Section 6.2). After that, you are ready to launch the analysis script; you can find detailed examples in the cookbooks of the analysis user manuals [1] – [4]. If the script ends with an error, you can check the reason of the failure on our error-code web-page
http://www.isdc.unige.ch/integral/analysis#Errors

To analyze the produced images it is convenient to use a program creating region files for DS9 from the produced catalogs (Section 6.3).

6.1 og_create

The first step in the data analysis is the creation of an Observation Group (OG) from the list of prepared Science Window Group DOLs you want to analyze. To create an Observation Group you use the *og_create* program. It also creates the standard **obs** branch of the directory tree, where the following analysis will be performed, and the high-level science window groups will be produced.

To follow one example given in the IBIS cookbook you create an ASCII file **dith.lst** with four lines in it:

```
scw/0044/004400540010.001/swg.fits
scw/0044/004400550010.001/swg.fits
scw/0044/004400560010.001/swg.fits
scw/0044/004400570010.001/swg.fits
```

This list can also look as simple as:

```
004400540010.001
004400550010.001
004400560010.001
004400570010.001
```

In Table 8 the parameters of the program are listed. Query parameters are marked in boldface.

Table 8: *og_create* parameters.

Name	Type	Description
		Parameters defining the name, contents and location of the OG
idxSwg	string	Index of science window groups or ASCII list of SWG DOLs, or just ASCII list of SWGs even without a path.
ogid	string	Observation group ID (used to build the path).
baseDir	string	Base directory from which the obs branch is built. This parameter should be a relative path from where *og_create* is run. If not defined, the executable will use the REP_BASE_PROD environment variable.
instrument	string	Comma separated list of instruments. This will create several OGs in the same repository.

		possible values: “INTEGRAL” → all data, “SPI”, “IBIS”, “JMX1”, “JMX2”, “OMC” → instrument specific data
Additional parameters		
obs_id	string	ISOC observation ID (copied to keyword OBS_ID in OG). (Optional)
scwVer	string	If the idxSwg is an ASCII list of just ScWIDs, this is appended to the end of the ScWID, unless there already is a version there. (Optional) default: “001” (as in: 001000010010.001)
purpose	string	Scientific purpose of the OG. (Optional)
swgName	string	The name that will be given to a science window group. (Optional) default: “swg” (as in: swg.fits)
versioning	integer	Add a version to the ogid directory (0 = NO, 1 = YES). default: 0

The command

```
og_create idxSwg=dith.lst ogid=dith baseDir="./" instrument=IBIS
```

creates a directory obs/dith with an Observation Group og_ibis.fits in it.

Note that if the file name passed as idxSwg includes a “+” or “[” sign, *og_create* automatically interprets that file as a FITS file.

6.2 Creating Good Time Intervals.

It may turn out that you want to analyze only a part of the time covered by your Observation Group (e.g., if you are analyzing a gamma ray burst). To analyze only part of the data you should define a Good Time Interval (GTI), which you will include into the analysis program as a user-defined GTI. For this purpose there is a very useful tool *gti_user*.

To create a GTI starting on IJD 1092.5 and lasting 1 minute (do not forget to convert it into days!) input the following command:

```
gti_user gti=user_gti.fits begin=1322.68 end= \
         length=6.944E-4 group=mygroup.fits\[1] unit="day"
```

This example uses an existing ISDC group mygroup.fits to get the overall TSTART and TSTOP.

This program also allows to generate several sequential GTIs in one go. The next example creates a list of 653 user GTIs of 1.5 seconds each, the start times of which are regularly spaced every 4 seconds. An overall observation start time is given at *tstart*=1850.0 (in IJD), and the first GTI should begin 3 seconds after this start time. The first GTI will thus cover the range [*tstart*+3.0 sec, *tstart*+ 4.5sec], the second [*tstart*+7.0 sec, *tstart*+ 8.5sec] and so on (note that *tstart* is given in IJD). The command is as follows:

```
gti_user tstart=1850.0 \
         unit=sec \
         begin=3.0 \
         end=4.5 \
         step=4.0 \
         repeat=653 \
         group=scw/027800480010.000/swg_jmx1.fits+1 \
         gti=user_gti.fits
```

The *unit* parameter specifies that the *begin*, *end*, and *step* parameters are given in seconds (default is IJD). Note that the duration of each GTI (1.5 sec in this example) is recorded in the difference *end-begin.* Note also that the *begin* and *end* parameters refer, in this example, to the start and end times of the *first* GTI in the series, given relative to the *tstart* value.

In that last example, you could drop the *group* option. In this case, no OBT would be computed (since the path to the auxiliary data necessary to compute the OBTs is in the group), and a warning message would be issued to inform you of this. But the created FITS file still contains the computed GTIs in both IJD and UTC formats, which can be used for your analysis. The full range of the possibilities of the program can be seen from the description of the parameters, given in Table 9.

Table 9: *gti_user* parameters.

Name	Type	Description
begin	string	Start time of a GTI. Can be given in UTC, IJD/MJD or in seconds from a start time (see `tstart` below). The recognition of the time format is done automatically: UTC times must be in the full UTC format YYYY-MM-DDThh:mm:ss. A purely numerical entry like "1234.5" will be interpreted as IJD if its value is below MJDREF=51544 and as MJD if above. It can also be the name of a file name preceded by "@". The format of this file is comment lines preceded by "#" and otherwise start and end times in IJD on the same line, separated by spaces.
end	string	End time of a GTI. See explanation for `begin` above for the possible time formats. The "@file" format is not recognized here. This parameter can be left empty if `length` is set instead.
length	real	Duration of a GTI in the defined time units (see `unit` below). If `end` is set this is ignored. <0: use end-begin. default: `-1.0`
repeat	integer	Number of repetions of defined GTI after the first. Useful, e.g., for phase-resolved analysis. The output table will contain the first GTI defined above and `repeat` replications shifted by `step` time units. default: `0`
step	real	Time interval between the start of one GTI and the next repetition in the defined time unit (see `unit` below). Only relevant if `repeat` is greater than zero. $< 0 =$ no repetitions default: `-1.0`
unit	string	Time unit used for `begin`, `end`, `length` and `step`. Possible values are "day" and "sec". If the former than `begin` and `end` are considered absolute values in IJD. If the latter, they are taken relative to `tstart`. default: "`day`"
type	string	Type of defined time interval: "good" or "bad". The latter can be useful to exclude certain times. default: "`good`"

tstart	real	Overall start time for GTI calculations. Relevant if a bad time interval is defined (then the time from `tstart` to `begin` is good time) or if `begin` is set relative. If left as a negative number, the value will be either read from the data structure addressed in `group` or set to a predefined fixed value default: `-1.0`
tstop	real	Overall end time for GTI calculations. Otherwise as `tstart`. < 0 =use default default: `-1.0`
group	string	DOL of a baseline group from which some values, especially `tstart` and `tstop` can be taken. In case of "inverse barycentered" GTIs (see parameter bary), this **must** be given and contain ISDC auxiliary data, especially orbital information matching the selected times.
bary	boolean	If set to yes, the user given GTI times are taken to be valid at the Solar System Barycenter. From these, the corresponding times of observation – needed for the GTIs – are then approximately derived. The accuracy of the inversion is 1ms. default: `no`
ra	real	The Right Ascencion of the source for which the barycentric corrections are calculated. Must be set to a reasonable value if bary=y. -99=`INDEF` default: `-99.0`
dec	real	The Declination of the source for which the barycentric corrections are calculated. Must be set to a reasonable value if bary=y. -99=`INDEF` default: `-99.0`
auxtype	integer	Selection flag which kind of auxiliary to use: 0 = any, i.e. use historic if possible, or snapshot or predicted 1 = use snapshot 2 = use predicted 3 = use NRT data (snapshot or predicted) 4 = use historic data default: `0`
tcor	integer	Required accuracy in time correlation: 0 = accurate, i.e. return NaN if no accurate correlation possible 1 = inaccurate, i.e. extrapolate time correlation forward if needed 3 = any, i.e. extrapolate time correlation forward & backward default: `3`
clobber	boolean	Shall existing output be deleted or not? default: `no`
chatter	integer	Verbosity level of the outputs (0 to 5 with 2 as "normal") default: `2`

6.3 cat2ds9

After analyzing your data with the ISDC scripts you have as a result an image file and a file with the names of the found sources. The *cat2ds9* program allows you to create a region file from your catalog, so that with DS9 you can see the position of the sources on the produced image. In Table 10 the parameters of *cat2ds9* are given.

The following two commands extract region files of the sources found in the course of IBIS data analysis and of the catalog sources.

Table 10: *cat2ds9* parameters.

Name	Type	Description
catDOL	string	DOL of catalog structure to convert to region file
fileName	string	Name of output region file
symbol	string	Symbol to use for source positions
color	string	Color for source symbol
font	string	Specify font to use for labels
ploterr	boolean	Plot circles with position error? default: `n`
raoff	real	Offset in R.A. for pure text labels default: `0.0`
decoff	real	Offset in Dec for pure text labels default: `0.5`

```
cat2ds9 ibis/res/isgri_mosa_res.fits\[2] found.reg symbol=box color=red
cat2ds9 ibis/res/isgri_catalog.fits\[1] cat.reg symbol=box color=white
```

6.4 mosaic_spec

mosaic_spec is a tool that extracts a count spectrum at a given sky position from a set of mosaic sky images. Intensity could be measured in a specific – or in the most significant – pixel, or derived from a gaussian fit with free or frozen position and width.

Note that if the significance of the point is less than five, then *mosaic_spec* will assign a non-zero value to its quality flag.

Table 11: *mosaic_spec* parameters.

Name	Type	Description
DOL_inp	string	DOL of the group containing the input images. default: “”
DOL_out	string	DOL of the group containing the output images. default: “”
EXTNAME	string	EXTNAME of the input images. default: “`ISGR-MOSA-IMA`”
DOL_idx	string	DOL of the index of input images. default: “”
DOL_spec	string	DOL of the index of output images. default: “”
ximg	real	Enter source coordinate x in pixels. possible values: `0 -- 10000` default: `0`
yimg	real	Enter source coordinate y in pixels. possible values: `0 -- 10000` default: `0`

ra	real	Enter source RA possible values: `-180 -- 360` default: `0`
dec	real	Enter source DEC possible values: `-90 -- 90` default: `0`
posmode	integer	Position fitting mode posible values: -1: freeze position as input; 0: position is always left free; 1: freeze position as found in first energy band default: `0`
widthmode	integer	Gaussian width fitting mode posible values: -1: radial width frozen as input 0: radial width left free 1: x and y width left free 2: x and y width fixed as found for first band default: `0`
psf	real	HWHM of PSF in arcmin possible values: `0.1 -- 20` default: `6`
size	integer	Enter source box half size in pixels on which the fit is performed. possible values: `1 -- 1000` default: `20`
back	boolean	Should a constant intensity background be considered in the fit default: `no`
allEnergies	boolean	Enter if the program has to select automatically energies default: `yes`
emin	string	Enter vector of energy bands minimum. default: “`25 30 40`”
emax	string	Enter vector of energy bands maximum. default: “`30 40 60`”
chatty	integer	Enter reporting-level (10: includes MINUIT log). possible values: `0 -- 10` default: `4`

References

[1] ISDC/OSA-UM-JEMX JEM-X Analysis User Manual.
http://www.isdc.unige.ch/integral/analysis#Documentation

[2] ISDC/OSA-UM-SPI SPI Analysis User Manual.
http://www.isdc.unige.ch/integral/analysis#Documentation

[3] ISDC/OSA-UM-IBIS IBIS Analysis User Manual.
http://www.isdc.unige.ch/integral/analysis#Documentation

[4] ISDC/OSA-UM-OMC OMC Analysis User Manual.
http://www.isdc.unige.ch/integral/analysis#Documentation

[5] ISDC/OSA–INST-GUIDE Installation Guide for the *INTEGRAL* Data Analysis System.
http://www.isdc.unige.ch/integral/analysis#Software

[6] CFITSIO Users Guide.
http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html

[7] A General Gamma-Ray Source Catalog by D.J. Macomb & N. Gehrels.
Astrophysical Journal Supplement, 120, 335 (1999)

[8] A catalogue of low-mass X-ray binaries by Liu, Q. Z.; van Paradijs, J.; van den Heuvel, E. P. J.
Astronomy and Astrophysics, v.368, p.1021-1054 (2001)

[9] A catalogue of high-mass X-ray binaries by Liu, Q. Z.; van Paradijs, J.; van den Heuvel, E. P. J.
Astronomy and Astrophysics Supplement, v.147, p.25-49 (2000)

[10] Catalogue of low-mass X-ray binaries by van Paradijs, J.
In "X-ray Binaries", Lewin W.H.G, van Paradijs J., van den Heuvel E.P.J., Eds., Cambridge University Press, Chapter 14, p. 536-577 (1995)

[11] The fourth UHURU catalog of X-ray sources by Forman, W.; Jones, C.; Cominsky, L.; Julien, P.; Murray, S.; Peters, G.; Tananbaum, H.; Giacconi, R
Astrophys. J., Suppl. Ser., 38, 357-412 (1978)