Using HIPS and MOC at the HEASARC

Tom McGlynn

NASA/HEASARC

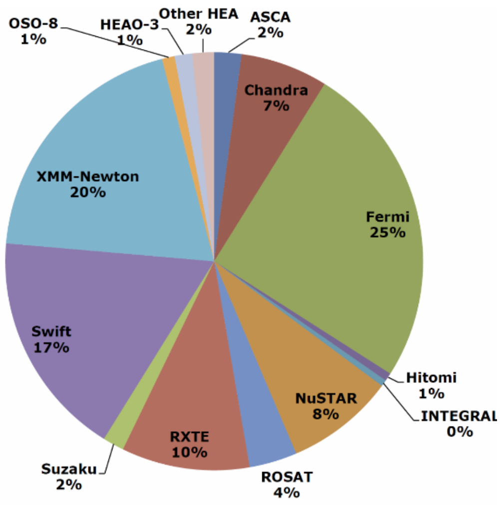
Summary

Using HiPS (with MOCs) to visualize and browse mission content

- Demos
  - SkyView
  - Xamin
- Issues with the HiPS and MOC standards
- General Recommendations for standards

HEASARC Background

- Founded 1990 as NASA's first domain archive
  - Gamma-ray, X-ray and more recently CMB datasets.
  - Over 60 missions including a number shared with other archives (CXC, ESA, JAXA, MPE) including Swift, Fermi, Chandra, XMM, INTEGRAL, Suzaku, Hitomi, WMAP, SPT, …
- Data often 'different' from the CCD paradigm common to typical IR/Optical/UV missions
  - Individual photons, soft boundaries to images, correlated pixels, Poisson statistics, exposures strong functions of position/energy, time variable sources, ….
- Only 100 TB but covering ~10 decades in energy (not including CMB). ~200 TB/year downloads
- Data universally correlated with other domains
- Already lots of cone, SIA, TAP, SSA services. HiPS is next!



HEA mission distribution of datasets retrieved from HEASARC

SkyView Image Access

- Create all sky summaries of Swift XRT and UVOT data
  - Counts and Exposure maps can be made with HipsGen
    - Some preprocessing of data required (e.g. split into simple FITS)
  - Intensity maps = counts/exposure using custom but simple code
  - About 300,000 total images, 3 TB/data
  - 2 months (including some reprocessing) using 2 VMs
- Add capabilities to *SkyView* to read HiPS locally and remotely.
- Add local and a few remote HiPS to *SkyView* queryable surveys (~40 total image planes)
  - Translate HiPS properties files to SkyView survey descriptions.

SkyView demo

Enable Visual Browsing of Archive Using AladinLite in Xamin

- Use HiPS generated for *SkyView*
- Create 3-color imagery using multiple filters in UVOT
  - Enables browsing multiple filters simultaneously
- Look for appropriate HiPS generated elsewhere.

- Add AladinLite to Xamin system

ADASS 2017 P145
Focus on general issues of integrating JavaScript libraries

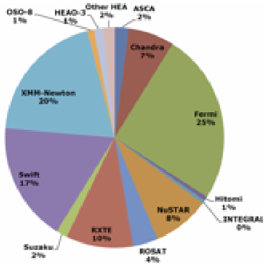
Using AladinLite and HiPS as a Visual Frontend in Traditional Archive Interfaces

Thomas McGlynn, Ben Pelletier, Alan Smale, Laura McDonald
NASA Goddard Space Flight Center

Issues and Plan

The High Energy Astrophysics Science Archive Research Center (HEASARC) serves as NASA's archive for over 35 high energy astronomy mission. Its Xamin system provides sophisticated capabilities for textual queries based on position, time, observational and object characteristics. However our existing system had limited capability enabling users to select regions visually or to display the locations of observations and targets against images of our archive holding. Creating such capabilities *ab initio* is a major software undertaking. Can we integrate an existing framework into our working system?

Goal: Integrate such a query and visualization into our existing systems using the AladinLite data visualizer developed at the CDS. Enable visualization of archive content by converting archive datasets to HiPS (Hiearchical Progressive Survey) using their HiPSGen tools. (See http://aladin.u-strasbg.fr/AladinLite and http://aladin.u-strasbg.fr/hips/ for details on these systems.)



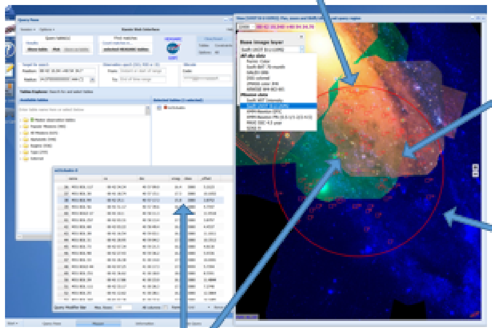
HEASARC datasets downloaded by mission (2016)

Most HEASARC science involves correlation of data from multiple missions so that understanding the spatial relationships among datasets is critical. The total download volume was roughly twice the archive size of 60 TB. Missions launched as recently as 2016 and as long ago as 1975 are named in this chart. These numbers do not include data retrieved through other archives, notably the CXC for Chandra or ESA for INTEGRAL and XMM-Newton among others.

Key Features

This simple example query looking for globular clusters in a region of M31 illustrates key features of the integrated system. Standard Xamin capabilities are used to select the table to be queried and add non-positional constraints, but AladinLite enables more graphic data selection and visualization.

Interactive Query Region Selection
Users can select a region to query by panning and zooming to some interesting region and then just doing a Shift/Click.



Visualization of Query Results
The spatial distribution of results is immediately shown and correlations – or not – with visible structures immediately manifest.

Browsing Archive Data
Users can browse archive holdings. Here we see the coverage of three different UVOT (B-U-UVM2) filters in the HEASARC archive. Regions with only two, one, or no filter are easily distinguished. In the bottom right only UVM2 data is available.

Cross-highlighting
Selecting an element in the table highlights the element in the visual display and vice-versa. Users can immediately determine the identity of interesting objects, or pick out objects with special non-spatial characteristics.

Status

In only a few months most of the key requirements for integrating visual browsing and selection have been met. Only very slight modifications were required to the AladinLite software itself, but the nominal, documented API was supplemented by looking within the JavaScript to access required capabilities that were available but not documented.

Significant enhancements to the HipsGen tools were needed to develop our HiPS datasets, however the HipsGen tools were crucial, with the supplements transforming the inputs into simple FITS and performing basic arithmetic on HiPS data generated by HipsGen (i.e., creating intensity maps by dividing total counts maps by total exposure maps). All public data from the Swift UVOT and XRT instruments were successfully combined into HiPS. The processing of the entire mission's datasets took two months.

Conclusions

The AladinLite mapper easily meets our basic requirements. It also allows users to visualize a given region in multiple wavelengths, to look for long term variability when we have multiple datasets in the same wavelength (e.g., Swift XRT versus XMM EPIC) and to view a variety of non-HEASARC datasets with the AladinLite browser.

While similar capabilities -- and more -- are available through other tools like the full-scale Aladin and the IRSA and MAST web interfaces, the ability to rapidly graft visualization tools into our existing web page suggests that well-designed JavaScript libraries and toolkits may be an effective way to minimize duplicate software development and maximize the science return for our community's software efforts. The cost of and time for implementing these features using AladinLite were dramatically lower than what we had estimated for an independent implementation.

Implications for JavaScript Libraries

Increasingly astronomy tools are using the browser as the application workspace with sophisticated applications being developed in JavaScript. This effort indicates that we can successfully integrate large, independently-developed JavaScript libraries to provide key functionality tied directly to the user interface.

We were successful even though the Xamin system is structured on top of the Sencha ExtJS library while AladinLite makes use of jQuery for its interactions with web elements.

When developing generic JavaScript capabilities, implementers should understand **and document** all aspects of the interface presented to the user.

- **The nominal API of the interface.** This includes global functions and name spaces defined by the libraries. To minimize collisions these should be kept to as small a surface as possible. For AladinLite, while the documented interface is limited to a single Aladin namespace, the actual code is somewhat promiscuous in consuming namespaces. While not a serious issue for us, this may cause problems as we begin to integrate many libraries into a single application. The Xamin system employs the ExtJS framework to bound itself to only a single namespace. *JavaScript libraries should minimize the namespace used.*
- **Redefining the prototypes of standard classes.** JavaScript allows code to redefine the meaning of fundamental objects, e.g., Strings, by adding new functionality or even replacing existing features. This is used minimally in AladinLite to facilitate JSON encoding. Any such behavior is dangerous in a multi-library environment since different classes may make incompatible changes. *Redefinition of standard classes should be avoided where possible.*
- **The page elements defined by the interface.** While some JavaScript libraries may be purely mathematical, the historical motivation for the language is the manipulation of web pages. The set of page elements that affect and are affected by a library is a key element of its interface. AladinLite packages all elements of the interface that are used within a single DIV. This minimizes interference with the other elements of the system. In practice this made it easy to integrate the AladinLite windows into Xamin. However the components within the Aladin DIV are accessible to the external code. These elements are also part of the implicit interface. *Elements used in the library should be clearly defined both visually and in the ID space used on the page.*
- **Events.** While the method calls in the API are the mechanism for invoking the included library, a carefully defined set of events can enable the library to notify the client of interesting changes but leave the client in charge of how these are used. AladinLite provides some capabilities here – notably with regard to selections of objects within catalogs – but other events like pans, coordinate system or resolution changes, or changes to the HiPS being examined are not made visible explicitly to the client. This is the only area where we felt we needed to make non-trivial modifications to the AladinLite code. *Use events to enable the client package to respond to changes in the library DOM space.*
- **StyleSheets.** The included library is likely to provide some level of CSS control with classes of settings that define the look of the elements controlled by the interface. These may conflict with the style of the other elements of the system. This is true for our AladinLite window where, e.g., the survey selection box is styled very differently from selection boxes elsewhere in the interface. Fortunately this window is dominated by the actual image graphics so that the textual elements that are most affected by the style setting are limited.

Future Plans

A beta version of our interface is available at the URL given below and will be released as our standard interface in November. We anticipate developing additional HiPS data for other missions (ROSAT, Suzaku, ASCA, Einstein), and we also may be using the VO MOC standard where it makes more sense to show coverage rather than actual image data.

Within Xamin we hope to extend our use of AladinLite to enable comparison of queries, blinking of survey data sets and other capabilities that enable the easy visualization and selection of HEASARC data.

Acknowledgements

We would like to thank Pierre Fernique in particular and the Centre de Donées astronomiques de Strasbourg (CDS) in general for their help in using HipsGen and AladinLite toolkits and for making these software resources available to the community. The external HiPS data used within our Xamin system includes HiPS generated by the CDS and JAXA as well as the HEASARC.

URL:
https://heasarc.gsfc.nasa.gov/xamin_test

Xamin demo

Issues with HiPS and MOC standards

Following notwithstanding, kudos to the HiPS and MOC designers for very useful standard.

- Audience for the standard are implementers. Can we make it easier for them?
- Where did we run into problems? Concentrate on a few points.
- More detailed comments in Word document also linked to talk. Includes purely editorial comments too.

Hiding HEALPix

Standards use external documents to define HEALPix

- Makes it harder for users who have to go elsewhere to understand HEALpix and original papers written with very different concerns than typical HiPS developer
- Easier if basic HEALPix transformations were documented internally – could be appendix
- Allows for clearer definition of meaning (e.g., doc has to discuss NESTED and RING without any context).
- Allows for easier comparison of new HEALPix ordering (which is one way of viewing HiPS) to older ones

Tiling trouble

- Key problem!
- What is the internal pixel ordering within a tile in various supported formats?
- Can this be affected by the WCS in FITS?
- In practice tried a variety of different possibilities until we got one that worked with Aladin.

Table Tension

- Structure of HiPS tables very unclear. Still don’t understand how this works.
- Do we repeat rows in deeper tiles? Standard suggests not but presume it must?

Advice acrimony

- HiPS is product not process.
- *Describe* the product don't *prescribe* how users build it, i.e., statements about how users generate deepest tiles or go from high to low order tiles.

More broadly for all VO Standards

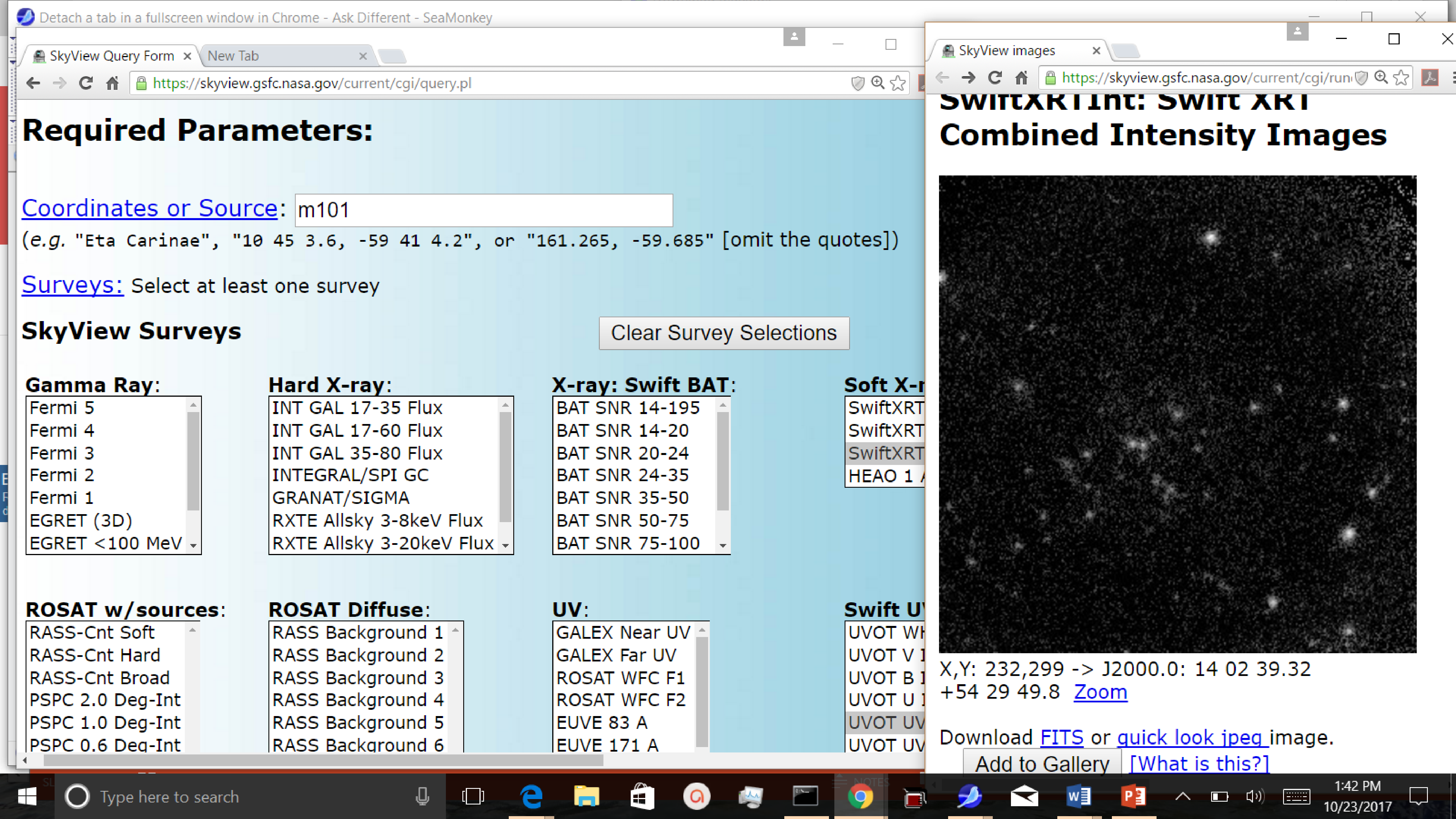
- Try to be self-contained.
- Be complete – implementer will not have background of writer
    - Complete worked examples showing details. Much easier to follow examples than prescriptions.
- Stay in scope
    - Clearly identify advisory statements and include them only when they will help implementer.

HiPS and MOC are great but we can strive to make IVOA standards clearer and more useful.

Summary

- HEASARC has successfully implemented HiPS standard and generated HiPS – with many thanks to CDS and Hipsgen tools.

- HiPS and MOC standards are nice but could be even better


Detach a tab in a fullscreen window in Chrome - Ask Different - SeaMonkey
SkyView Query Form
New Tab
https://skyview.gsfc.nasa.gov/current/cgi/query.pl
Required Parameters:
Coordinates or Source: m101
(e.g. "Eta Carinae", "10 45 3.6, -59 41 4.2", or "161.265, -59.685" [omit the quotes])
Surveys: Select at least one survey
SkyView Surveys
Clear Survey Selections
Gamma Ray:
Fermi 5
Fermi 4
Fermi 3
Fermi 2
Fermi 1
EGRET (3D)
EGRET <100 MeV
Hard X-ray:
INT GAL 17-35 Flux
INT GAL 17-60 Flux
INT GAL 35-80 Flux
INTEGRAL/SPI GC
GRANAT/SIGMA
RXTE Allsky 3-8keV Flux
RXTE Allsky 3-20keV Flux
X-ray: Swift BAT:
BAT SNR 14-195
BAT SNR 14-20
BAT SNR 20-24
BAT SNR 24-35
BAT SNR 35-50
BAT SNR 50-75
BAT SNR 75-100
ROSAT w/sources:
RASS-Cnt Soft
RASS-Cnt Hard
RASS-Cnt Broad
PSPC 2.0 Deg-Int
PSPC 1.0 Deg-Int
PSPC 0.6 Deg-Int
ROSAT Diffuse:
RASS Background 1
RASS Background 2
RASS Background 3
RASS Background 4
RASS Background 5
RASS Background 6
UV:
GALEX Near UV
GALEX Far UV
ROSAT WFC F1
ROSAT WFC F2
EUVE 83 A
EUVE 171 A
SkyView images
https://skyview.gsfc.nasa.gov/current/cgi/run
Combined Intensity Images
X,Y: 232,299 -> J2000.0: 14 02 39.32 +54 29 49.8 Zoom
Download FITS or quick look jpeg image.
Add to Gallery [What is this?]
Type here to search
1:42 PM
10/23/2017


Xamin Web Interface
https://heasarc.gsfc.nasa.gov/xamin_test/
Query Pane
Session
Options
Xamin Web Interface
Help
Query table[s]
Results:
Show table
Plot
Save as table
Find matches
Count matches in...
selected HEASARC tables
HEASARC
GSFC
Clear/Reset ...
Tables
Constraints
Options
All
Target for search
Position: 00 45 26.11 +40 59 13.8
Radius: 80.67333333333
min (')
Observation epoch (ISO, MJD or JD)
From: Instant or start of range
To: End of time range
Bibcode
Code: YYYYjjjjjvvvvsppppA
Tables Explorer: Search for and select tables
Available tables
Enter table name here or select below
Master observation tables
Popular Missions (480)
All Missions (625)
Alphabetic (948)
Regime (936)
Type (259)
External
Selected tables (1 selected)
messier
View [UVOT B-U-UVM2]: Pan, zoom and Shift/click to set query region
J2000
00 45 26.110 +40 59 13.80
messier 0
	name	alt_name	ra	dec	constell	dimension	vmag	vmag_uncert	class	_offset
1	M 32	NGC 221	00 42 42.0	40 52 00	AND	8X6	8.2	null	6300	31.8384
2	M 31	NGC 224	00 42 42.0	41 16 00	AND	178X63	3.4	null	6200	35.1657
3	M 110	NGC 205	00 40 24.0	41 41 00	AND	17X10	8.0	null	6300	70.4354