*FULL VOLUME TITLE*
*ASP Conference Series, Vol. *, © 2007*
*Editor 1, Editor 2, and Editor 3, eds.*

# Chapter 1: Using Virtual Observatory Services in *SkyView*

Thomas A. McGlynn

## Introduction

For over a decade *SkyView* has provided astronomers and the public with easy access to survey and imaging data from all wavelength regimes. *SkyView* has pioneered many of the concepts that underlie the Virtual Observatory. Recently *SkyView* has been released as a distributable package which uses VO protocols to access image and catalog services. This chapter describes how to use the *SkyView* as a local service and how to customize it to access additional VO services and local data.

## 1. Getting Started.

The geometry of astronomical images is complex. There are dozens of projections, tangent, Cartesian, Aitoff, ..., that can be used to translate from the celestial sphere to a plane, There are coordinate systems, Galactic, equatorial, ecliptic, ... and different equinox epochs for some choices. The position orientation and scale of images can differ. Data may all be provided in a single image, or broken up into a mosaic where images may or may not overlap—and whose internal geometries may or may not be consistent. Over a decade ago, the *SkyView* virtual telescope was built so that astromomers could generate images of any region of the sky in any wavelength without having to worry about these details. *SkyView* took care of the geometry, and let the astronomer concentrate on the science. Recently the *SkyView* geometry engine has been released as a distributable package. Users can run *SkyView* locally and customized it to their particular needs, adding local data, or adding new survey datasets or catalog overlays using Virtual Observatory protocols.

This chapter describes how you can use *SkyView* out of the box to retrieve data from dozens of survey datasets. It then discusses how *SkyView* can be customized and how easy it is to use SkyView to extract information from new data sources so long as they are accessible through VO protocols.

*SkyView* is not primarily a Virtual Observatory service. Rather, it shows how a service can use the VO to get expand its reach and flexibility. The *SkyView* distribution is included in the CD, and is available on the Web in the NVO Summer School distribution or the *SkyView* Web site at http://skyview.gsfc.nasa.gov/jar/jar.html.

To use the *SkyView* distribution you should get the *SkyView* JAR file from any of these sources and place it somewhere on the machine where you wish to run it. You will also need a version of Java at least as recent as Java 1.5. All of the software, data and configuration files needed to execute *SkyView* are included in the JAR file. The *SkyView* web site contains complete documentation of the distribution and includes descriptions of over 50 surveys available and the myriad options available for data processing. The goal of this chapter is to show how easy it is to get started using *SkyView*.

## 2. Using Built-In Surveys

Let's jump in. To generate an SDSS image of the galaxy M87 enter the command

```
    java -jar skyview.jar survey=sdssi position=m87
```

This assumes the skyview.jar is in the current directory. If not just use its path. You'll also need to be on the Web. There are a few surveys that are entirely contained within the JAR file, but most use resources on the Web, and in any case we need to access NED and SIMBAD to translate the name m87 to coordinates.

This generates the output

```
Processing survey:sdssi
Number of candidate source images is 2.
Processing image #1
Processing image #0
Opening output file output.fits.
```

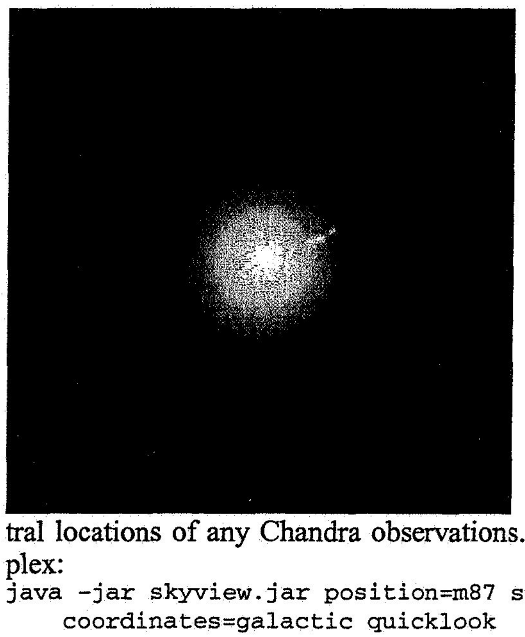
What has happened here? *SkyView* first goes to an SDSS Virtual Observatory Service (a Simple Image Access service see [?? in this volume]) and finds all of the I-band images it has that might overlap the region around M87. The SDSS service returns two candidate images. The program then looks at each pixel in the output image and selects the best of the input candidates to use to sample for that image. It then downloads the images it needs from the SDSS, resamples the output coordinate grid and generates a FITS image.

Many details have been defaulted: the coordinate system defaults to equatorial, the projection to a Tangent plane/gnomonic projection, the scale to the scale of the survey pixesls, .... *SkyView* makes reasonable choices for all of these; the defaults can be set on a survey by survey basis.

We can reenter the first command as

```
    java -jar skyview.jar survey=sdssi position=m87 quicklook=jpeg
```

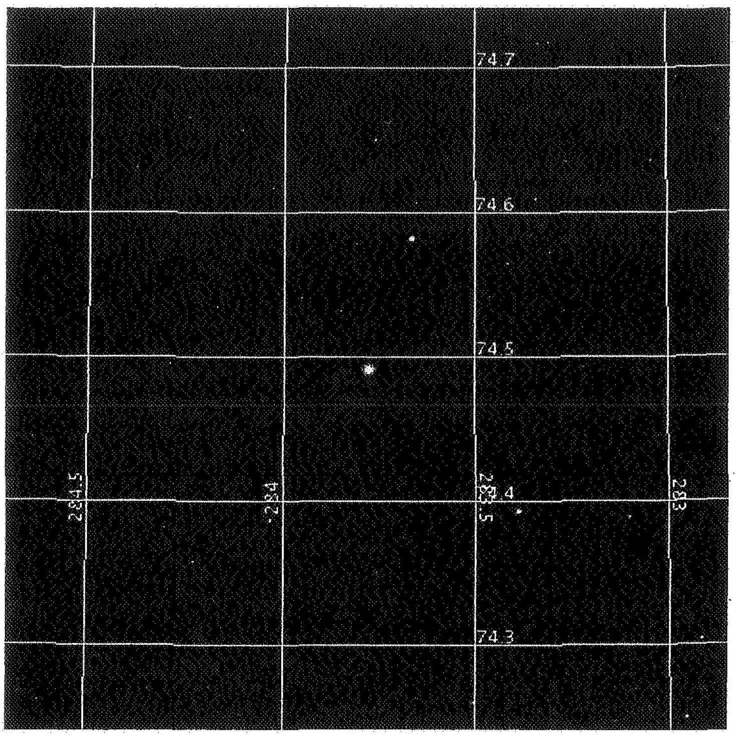
Now in addition to the FITS file a JPEG quicklook image will also be generated.



We can see the structure in the M87 jet easily. There a lots of options to control the color table and scaling of the image. By default you get a 300x300 pixel image with a scale very similar to the original survey data (1" pixels). You may wish a larger region in the sky or to get a larger number of pixels. You can also request that the image be overlaid with a coordinate grid and to indicate where catalog objects would fall in the image.

Suppose you want a 500x500 pixel image of the 30' region around M87 in galactic coordinates with a coordinate grid and you also want to know the central locations of any Chandra observations. The command gets a little more complex:

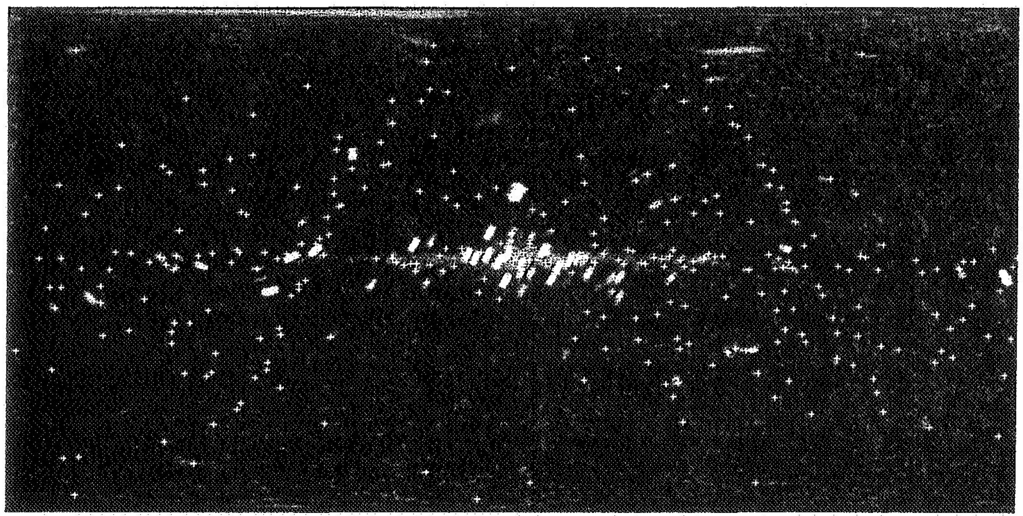
```
java -jar skyview.jar position=m87 survey=sdssi size=0.5 pixels=500
    coordinates=galactic quicklook
```



There are over fifty surveys packaged in the distribution ranging from radio through gamma-rays including the SDSS, 2MASS, DSS, FIRST, and other surveys and many capabilities for customizing the result. Users can select the coordinate system, epoch, scale orientation and sampling to be used for images. There are also facilities for matching edges when mosaicking data from surveys where the backgound may not be consistent from image to image. For most of the large datasets data is retrieved using the VO Simple Image Access Protocol.

Catalog positions may be overlaid on the maps. E.g.,

```
java -jar skyview.jar survey=heao1a position=0,0 size=360,180
   pixels=800,400 quicklook catalog=uhuru4 coordinates=galactic
```



creates an all sky overlay of the positions of the Uhuru X-ray catalog on an all-sky HEAO 1A X-ray survey. Each catalog object is marked with a cross. Catalog information is retrieved using the VO Cone Search protocol. Tables available at the

Vizier or at the HEASARC can be retrieved by name, or the user can specify the URL of the cone-search service in as the value of the `catalog` parameter.

There are lots of parameters and you can create a settings file to set up the defaults you want to use.

## 2. Adding Surveys and Catalogs.

Users can easily add new surveys and catalogs to SkyView to link to new VO services or their own local data. While its perfectly feasible to make links to data that is retrieved through some custom interface, it is especially easy if the data is provided by the Simple Image Access protocol. Each survey is described by a Survey Description file, a simple XML file that gives the metadata about the survey and information about the files contained in it. Let's take a look at the description of the SDSS I survey we used above in some of the example above and indicate how you might modify it to add some new survey. This file is included in the SkyView JAR. The shaded elements are from the XML file. Comments and some documentation lines are removed for brevity.

```
<Survey>
  <ShortName>SDSSi</ShortName>
```

The short name is used to define the survey you when you run invoke SkyView. Make it some short memorable string (matching is not case sensitive).

```
<Name> Sloan Digitzed Sky Survey I-band </Name>
  <Description>
...
  </Description>
```

The Name and Description fields describe the survey. They are not used by the program and are optional.

```
  <Settings>
    <Scale> 0.00011 </Scale>
  </Settings>
```

Settings are used to specify survey specific defaults. Usually you want to specify a default size for pixels. The scale is in degrees, so this survey has pixels just under 0.5".

```
  <MetaTable>
...
  </MetaTable>
```

The MetaTable parameters give some standard metadata information about surveys. These are optional.

```
  <Images>
```

The images area is where we define what we know about the image geometry.

```
    <SiapURL>
<![CDATA[
http://casjobs.sdss.org/vo/DR5SIAP/SIAP.asmx/getSiapInfo?FORMAT=image/
fits&BANDPASS=i&
]]>
    </SiapURL>
```

This URL is the base URL for the SIA service. You should substitute your own.

```
    <SiapProjection>  Tan   </SiapProjection>
    <SiapCoordinates> J2000 </SiapCoordinates>
```

These two fields give information about the geometry of the images returned by this SIAP service. Typically an SIA service gives back reasonably uniform images. The geometry fields that are constant from image to image are specified here. Other fields that might be specified if they are not given in the SiapNaxis and Scaling to give the number and size of pixels in the image. Usually these two are given in the SIA service.

```
    <ImageFactory>
         skyview.survey.CachingImageFactory
    </ImageFactory>
```

This class is what SkyView uses to get files from remote locations. You shouldn't need to change it unless you have special requirements.

```
    <ImageSize> 0.25 </ImageSize>
```

The image size is the typical size of images in degrees. Change it to an appropriate size for your survey.

```
    <ImageGenerator>
       skyview.survey.SIAPGenerator
    </ImageGenerator>
```

This indicates that you want to find candidate images using the VO SIAP protocol.

```
  </Images>
</Survey>
```

To write a service description to incorporate an SIA service into *SkyView* all you need to know is the base URL for the service, and the basic geometry of the images the service returns (e.g., the coordinate system, and sizes).

Not all SIA services may be appropriate for inclusion into *SkyView*. If a service returns sets of images where it doesn't really make sense to mosaic them together, e.g., some X-ray, some optical, some radio images, then be careful. However it may be possible to filter the images to get just one set of images that is appropriate for mosaicking. Some heterogeneity is OK. E.g., images need not have exactly the same size or resolution, but large variations may be difficult to handle.

Sometimes you may wish to add local data as a survey without making an SIA service. You can create a descriptor file that explicitly names all the files in your survey, or you can use the UserFile argument to the *SkyView* command line. These options are described in the *SkyView* documentation.

Getting catalog overlays is more straightforward. If there is a VO cone search service, you can invoke it by specifying `catalog=url` where the url is the base URL for the cone search. E.g.,

catalog=http://myconesearch.stateu.edu/cgi-bin/vo/consearch. If you want more than one, specify multiple URLs separated by commas.