

# **Astro-Wise Image Services**

Astro-Wise Workshop Nov 2006

# AWE-Image client and server

## Overview:

- AWE-image-server services and operation
- Interaction with web interface (DBview)
- Interaction with client from AWE or python prompt
- Output from image-server
- Screen shots

# AWE-Image-data web-services

## Mode of operation:

- Single AWE-Image-server is connected to multiple local data-servers storing FITS-images
- Image-Operations on-demand :
  - Web-browser-viewable pictures of entire images
  - Cut-out sub-images to be viewed in Web-browser or to be downloaded as FITS
- Experimental issues:
  - response-time and storage-space requirements
  - efficient output formats

# Image-processing operations

External programs are used for image-processing:

- Producing sub-images [cut-outs] (fitscopy)
- Conversion of fits-format to tif-format (stiff)
- Conversion of tif to png and jpg for web-viewing (convert)
- Compression and combination of sub-images (zip)
- Overplotting with textlabels,positions: (convert)

# Interaction with DBview-web-interface

- Image-Services from DBview- web -interface
- rendition of all images in database :
  - optimized for speed and size
  - basic control over contrast, size, color-reversal

# Rendition of FITS-images via DBview-webinterface

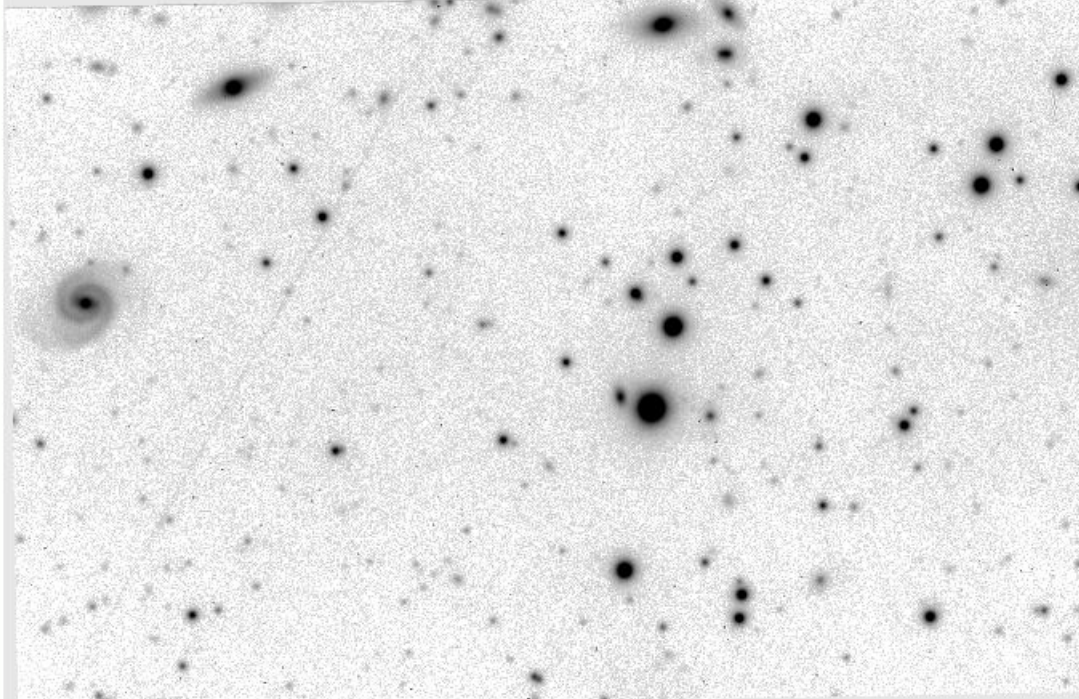
- Variable binning (1..8) to fit images onscreen
  - for more details use cut-out service or external program
- Automatic setting of contrast

# Viewing whole images

Call an image from a button in the table for on-demand processing:

<http://imageview.astro-wise.org:8990> - Sci-TSCHNEIDER-MDM8

[POS image](#) [CONTRAST HIGH](#) [CONTRAST LOW](#) [Resize 0.5x](#) [Resize 2x](#) binningfactor = 4 [\[RE\]](#)



messages: ----

Done

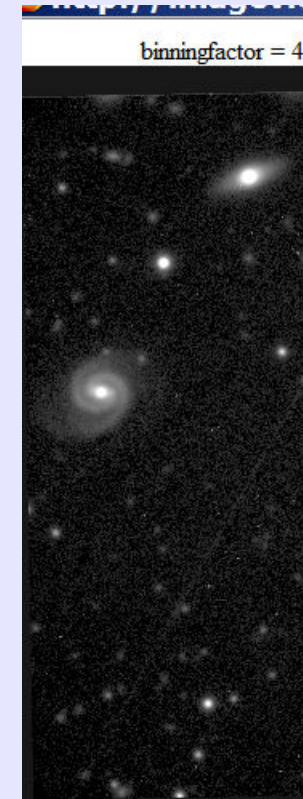
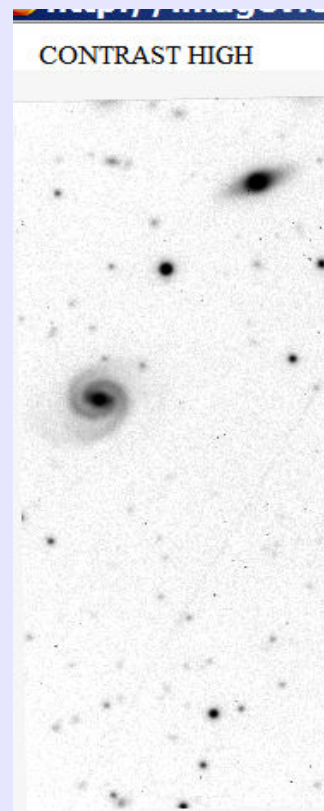
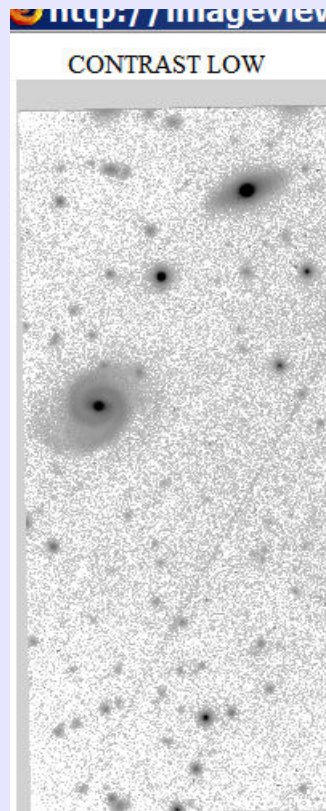
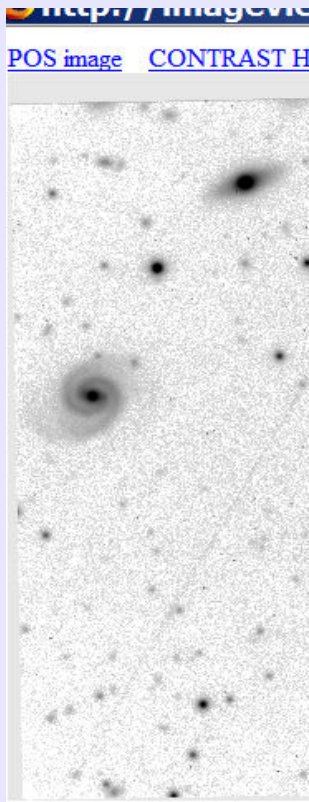
## RegriddedFrame

EROPNT	ZPNTERR	creation_date	filename	globalname
3481462192	0.00150280040095	2006-10-20 17:40:18	<a href="#">Sci-TSCHNEIDER-MDM8K-----R-4-Regr--Sci-54028.7361976-387926317c81c90b05986445ef19c66045a531a8.fits</a> <a href="#">Picture</a>	NULL
3632368367	0.000696019106664	2006-10-20 17:40:15	<a href="#">Sci-TSCHNEIDER-MDM8K-----R-5-Regr--Sci-54028.7361941-1e06bfa7c4b1801fb31b02915545a45e8294607.fits</a> <a href="#">Picture</a>	NULL
5131260679	0.00121200868819	2006-10-20 17:40:15	<a href="#">Sci-TSCHNEIDER-MDM8K-----R-3-Regr--Sci-54028.7362034-4488b43dd72bf3d87be6529c005d4ee9b48f09e2.fits</a> <a href="#">Picture</a>	NULL
3709113753	0.000805183945091	2006-10-20 17:40:14	<a href="#">Sci-TSCHNEIDER-MDM8K-----R-6-Regr--Sci-54028.7361992-c8eff3ed3907fa696c890662ac423a715287cfd.fits</a> <a href="#">Picture</a>	NULL

McAfee SiteAdvisor

# Tweak the image

Change contrast or invert color



# Obtaining cut-out images from the AWE- prompt

- Data-input to python image-client:
  - Dictionary with file and positional data
- Use available functions/methods to make input:  
Applies to SourceList and AssociateList classes
- Generate input by hand or own method
- Select output-format : FITS-files, or png-images  
linked from an HTML-table

NB HOWTO is available from [portal/manual](#)

# Object methods exporting image-data

Some Objects have methods to return data to get cut-out-FITS-images from the image server:

- `SourceList : .sources.make_image_dict(sids, mode='sky')`
- `AssociateList: associates.make_image_dict(aids, mode='sky')`

sids/aids may contain one or a list of SID/AID.

“Mode” can specify sky (default) or grid coordinates.

# Input Example 1: SourceList-generated data

Import program modules:

- `awe> from Services.imageview import imgclient # interface to image-server`
- `awe>from astro.main.SourceList import SourceList`

Get data for cutouts of SID # 1..10 :

- `awe>sourcelist = (SourceList.SLID == 54052)[0] # any query`
- `awe>imgdict=sourcelist.sources.make_image_dict(range(10),mode='grid')`

Send data to the image-server with size-option:

- `awe>ic = imgclient.imgclient(imgdict, wide_high=[150, 150]) # set size-option`
- `awe>ic.getzipfile() # receive zipped cut-out fitsfiles`
- `awe>ic.getimg() #receive an HTML-page with image-links`

# Input Example 2: user-generated data

## Construct user-dictionaries {filename : pdata}:

- Lists with optional height and width parameters per individual sub-image:
  - Pdata = [['11.5557568095', '-29.3887976732'], ['11.4245138174', '-29.3883174776',"200","200"], ['11.5253142382', '-29.3874586034',300,300]]
- -Dictionaries specifying position, size and additional items:
  - Pdata= [{'RA':'11.5557568095', 'DEC':'-29.3887976732',"extra1": 123,"extra2": 321}, {'RA':'11.4245138174', 'DEC':'-29.3883174776',"width":"200","height": "200"}, {'RA':'11.5253142382', 'DEC':'-29.3874586034',"width" : 300, "height": 300,"extra1": 123,"extra2": 321,"extra3": 567,"extra4": 8910}]

## Special dictionary keys are:

- PIX\_X,PIX\_Y instead of RA,DEC: to enter pixel-coordinates
- width,height : individual sub-FITS-image dimensions in pixels
- NB see wide\_high parameter above for an overall size-setting
- FTAG: for a user-defined tag added to the output-filenames
- Additional items are placed into the FITS-headers.

**Instantiate imgclient, request images as in Example 1**

# Response from image-server -1

After instantiating the image-client, the input and the presence of requested files on image-server are checked.

Messages seen from the prompt (example):

**start imgclient**

**inittest**

**Dictionary object-length 1855**

**Number of base-image names 1**

**Number of sub-image positions 49**

**file GSIKK844Coadd.fits present on dataserver**

**Will POST 3736 bytes to [imageview.astro-wise.org:8990](http://imageview.astro-wise.org:8990)**

# Response from image-server -2

After processing `ic.getimg()`:

**HTTP RESPONSE: 200**

**Received from server [imageview.astro-wise.org:8990](http://imageview.astro-wise.org:8990): HTML links/table  
30084 bytes.**

**Stored in current directory: file `Md6guxo.html`**

**Files are present on the image-server and can be reached from HTML-links.**

After processing `ic.getzipfile()`:

**HTTP RESPONSE: 200**

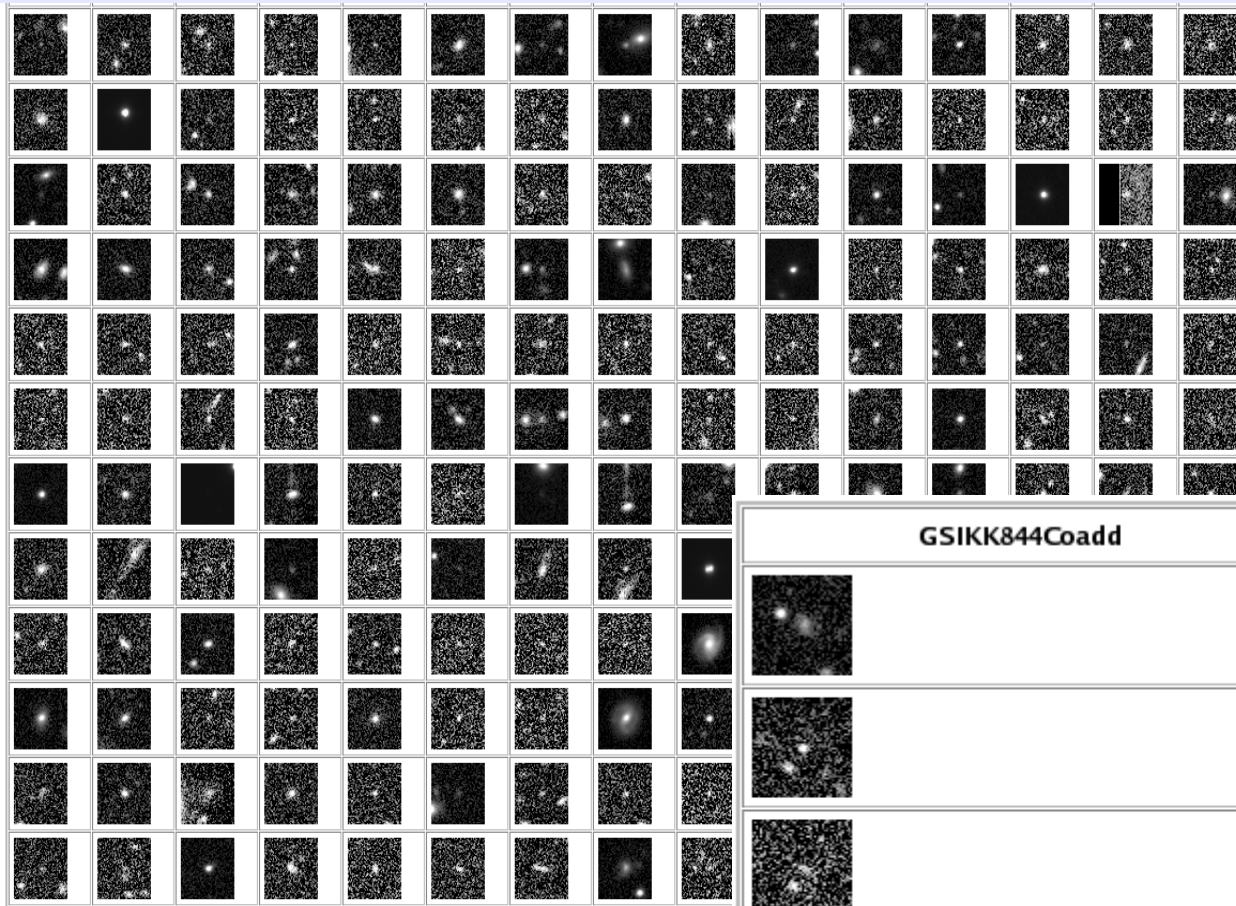
**Stored in current directory: file `GSIKK844Coadd__Fkb6.subALL.fits.zip`**



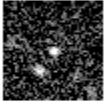





**Sub-image FITS-files are available on the clients machine in a zip-archive.**

# View and compare image-server output via HTML-windows and tables

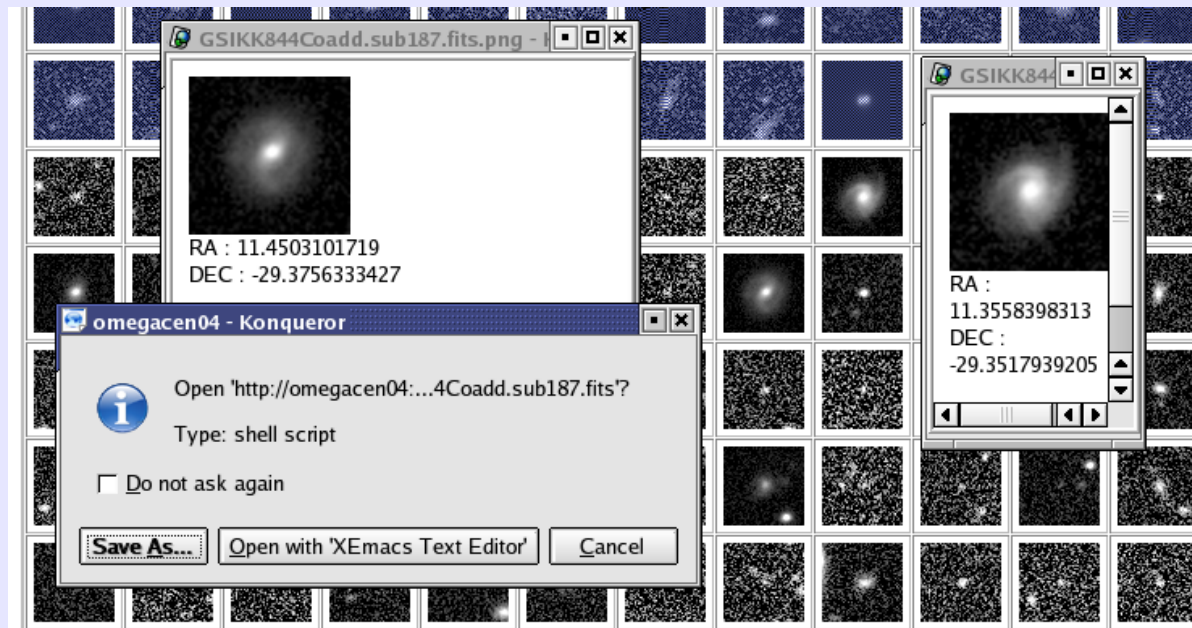
- In tables:
  - Sub-images from a single base-image placed in 16 x 30 tables
  - Sub-images from multiple base-images arranged in separate columns of a table.
- In individual windows:
  - Single sub-images enlarged into a private window: move anywhere on the screen for comparisons within and between tables.

# HTML-picture-link tables



GSIKK844Coadd	GSIKK842Coadd
	
	
	
	

# single-sub-image magnification, rearranging and download



# Directory-links to view or download

All files involved can be seen and downloaded from an HTML-sub-directory-view

<a href="#">GSIKK842Coadd.sub6.fits.png</a>	<a href="#">GSIKK844Coadd.sub2.fits.png</a>	<a href="#">GSIKK844Coadd.sub2.fits</a>
<a href="#">base_imgslst.py</a>	<a href="#">subimgslst.py</a>	<a href="#">GSIKK842Coadd.sub6.fits</a>
<a href="#">GSIKK844Coadd.sub1.fits.png</a>	<a href="#">GSIKK842Coadd.sub7.fits</a>	<a href="#">GSIKK844Coadd.sub4.fits</a>
<a href="#">GSIKK844Coadd.sub0.fits.png</a>	<a href="#">GSIKK842CoaddbBO1zH.subALL105.fits.zip</a>	<a href="#">GSIKK844Coadd.sub4.fits.png</a>
<a href="#">GSIKK842Coadd.sub5.fits.png</a>	<a href="#">GSIKK844Coadd.sub1.fits</a>	<a href="#">GSIKK842Coadd.sub5.fits</a>
<a href="#">GSIKK844Coadd.sub3.fits.png</a>	<a href="#">GSIKK844Coadd.sub3.fits</a>	<a href="#">GSIKK842Coadd.sub7.fits.png</a>
<a href="#">GSIKK844Coadd.sub0.fits</a>	<a href="#">REDUCED LARGE IMAGE GSIKK844Coadd.fits</a>	<a href="#">REDUCED LARGE IMAGE GSIKK842Coadd.fits</a>

[\[\\* .png \]](#) [\[\\*x2.jpg\]](#) [\[\\* .zip\]](#) [\[\\* .fits\]](#) [\[\\*.\\* : ALL\]](#) [\[CLEAR files: \\*.\\*\]](#)

# Image-server and client: status and development

- Primary AWE-Image services are available from Web-interface and AWE prompt: Whole images and cutouts can be obtained
- Plans for Additional image-services are:
  - Methods to obtain series of related images
  - Web-interface for cut-out images
  - Overplotting positions from SourceLists/catalogs