

# Accessing XXL thumbnails and catalogues

## A java standalone tool

L.Chiappetti<sup>1</sup>

INAF, IASF Milano, via Bassini 15, I-20133 Milano, Italy

**Abstract.** I report on a standalone ad-hoc java tool which will allow to access the optical, NIR, IR and UV thumbnail images around XXL X-ray source positions together with the tentative identifications recorded in the database. It is hoped that this tool will overcome the difficulties some people had to access previous (applet or JWS) tools developed for XMM-LSS.

**Key words:** LSS; XXL; data products

### 1. Introduction

In the framework of last April AGN meeting in Athens, there was expression of increased interest in accessing thumbnail optical images around XXL X-ray sources. This interest was confirmed at the June XXL Consortium meeting in Sesto. I decided therefore to tackle the problem, building up on previous experience done in XMDS and XMM-LSS.

There are at least three separate items which interlace when we talk about thumbnails:

- The first one is the **generation of thumbnails** themselves and their storage within the XXL database.
- The second item is the **reference multiwave catalogue** one uses to tell which object in the thumbnail image is associated with which source in the various non-X-ray bands.
- The last (and not least) item is the **visualization tool** one may use to perform the inspection described in the previous sentence.

For what concerns the **generation of thumbnails** the task has been taken by Sotiria Fotopoulou, which had already extracted the cutout images for her own use, and kindly supplied them to me after Sesto. The thumbnail `fits.gz` images are now available for retrieval as data products from the XXL database as explained in section 1.1. This way anybody can bring *any* of them home and use them there with the tools one likes.

For what concerns the **reference multiwave catalogue**, we do not yet have an official one, not even

for internal use, unlike pure X-rays where we have the `XXLN` and `XXLS` described in Report XII (Chiappetti, 2013). Unofficially I provided two multiwave catalogues within the database, called `XXLNOPT` and `XXLSOPT` done "my way" (i.e. analogous to the published `2XLSS` catalogue) and using the non-X-ray catalogues listed in Report XII. In the framework of the activity solicited and lead by Geneva I also generated analogue associations using the non-X-ray *homogenized* catalogues prepared by Sotiria Fotopoulou. All this is described in Report XIV (Chiappetti, 2014). Database-wise they aren't immediately available (there are the GCTs `testxon` and `testxos` underlying them, but no queryable `VIEWS`). However for what concerns the tool described here one can access the list of associated sources under the names of `TESTXON` and `TESTXOS` (capital). This is felt to be enough to give anyhow prompt access to data though unofficial, and not stop people from working.

For what concerns the **visualization tool** a *Java applet* has been in use in Milan already for XMDS, and was adapted to XMM-LSS and somehow used also in Liege. However people had difficulty accessing it (and also the more recent *Java Web Start* version). User interested in the history of such older version can consult section 1.2. This report presents a new Java standalone application which users can download and run on their machine, accessing thumbnails and multiwave catalogues and hopefully overcoming the access difficulties encountered so far.

This report is arranged as follows: section 1.1 provides information about retrieving the data products from the database and bringing them home (consistently with the e-mail announcement also stored on the wiki); sections 1.2 and 4 will provide links to historical and technical material available as web pages; section 2 will provide a quick start guide to retrieving, testing and using the new visualization tool, while section 3 will provide a command reference in an easier way than what sparse in older web pages.

Typical users will start from 2 and read also 1.1 and 3; the rest is just for the curious.

### 1.1. Data products

Thumbnail images ( $30'' \times 30''$ ) have been cut out by a number of optical, NIR, IR and UV surveys by Sotiria Fotopoulou (SF). Please address to her for any specific query.

Such images are stored as *database data products* as *gzipped FITS* images. There is a directory (referred to as *repository*) per X-ray source, identified by its `Xseq`, and containing images from various catalogues and bands with names like:

`Xseq_catalogue_survey_band.fits.gz`  
e.g. `200002_GALEX_DIS_NUV.fits.gz`

The presence of catalogues and bands per source depends on availability (inquire with SF). Possible catalogues and bands are listed in Table 1. Not all of them are necessarily available, and in particular not all of them are made available via the visualization tool (which picks up the "preferred band"). The only way to get ALL available bands is to retrieve the images as data products.

From the point of view of database the data products are *nominally* associated to tables `XXLNOPT` and `XXLSOPT`. However this does *not* imply any relation of the images to the sources in the database non-X-ray table, nor to the associations done by various people and supplied to Geneva.

To retrieve the data products one proceeds as follows:

- log in to the XXL database in workspace `xx1`
- in the `Tables` pane web page, tick either `XXLNOPT` or `XXLSOPT`.
- *do not forget* to tick also "Show member tables (and data products!) also" which appears at the very bottom of the table list!
- perform any query
- then in the "Query results" page, click on "View data"
- click on the last column labelled "Object related files" (if the column is systematically empty you forgot to tick the "data products also" tick box!)
- open the branch labelled "thumbnail repository" then the folder corresponding to each `Xseq` of interest, as shown in Fig. 1 and retrieve the wished files
- alternatively from "Query results", click on "Retrieve all object related files"
- open the branch labelled "all thumbnails", as shown in Fig. 2
- clicking on "online generation" will allow downloading of a `tar.gz` file containing *all* the thumbnails of *all* sources returned by the query

*Please use sparingly the facility producing a `tar.gz` file with all thumbnails (i.e. limit yourself to a few sources at a time) in order not to saturate the system: the entire data product set consists of > 14000 directories in the north, and > 11000 in the south, each one with several images!* There is a quota system, so you can produce a

Catalogue/survey	bands	availability
<b>North<sup>†</sup></b>		
CFHTLS_D1	ugriyz	few
CFHTLS_W1	ugriyz	almost all
CFHTLS_WA	grz	few
CFHTLS_WB	grz	few
CFHTLS_WC	g	few
GALEX_AIS	NUV FUV	very few
GALEX_DIS	NUV FUV	almost all NUV
GALEX_GII	NUV	few
GALEX_MIS	NUV	very few
IRAC	ch1 ch2	almost all
SDSS	ugriz	almost all
UDS	BVRiz	few
UKIDSS_DXS	JK	few
UKIDSS_UDS	JK	very few
VHS	H J $K_s$	almost all
VIDEO	HJK <sub>z</sub>	few
VIKING	HJK <sub>s</sub> YZ	some
WIRcam	K	many
WISE	W1 W2 W3 W4	many
<b>South</b>		
BCS	griz	almost all
DECam	griz	almost all gz
GALEX_AIS	NUV FUV	few
GALEX_GII	NUV FUV	few
GALEX_MIS	NUV	very few
IRAC	ch1 ch2	nearly all
VHS	H J $K_s$	almost all
WISE	W1 W2 W3 W4	almost all

**Table 1.** Presently available catalogues, surveys and bands with thumbnails

<sup>†</sup> In addition there are 27 cases labelled `CFHTLS_W_u.020241-050800_T0007_MEDIAN.fits` and 327 cases labelled `CFHTLS_W_u.020631-041200_T0007_MEDIAN.fits`.

tar file over quota, but have to delete it before producing more. However if the tar file is too big, the performance of the database server cannot be predicted.

### 1.2. History

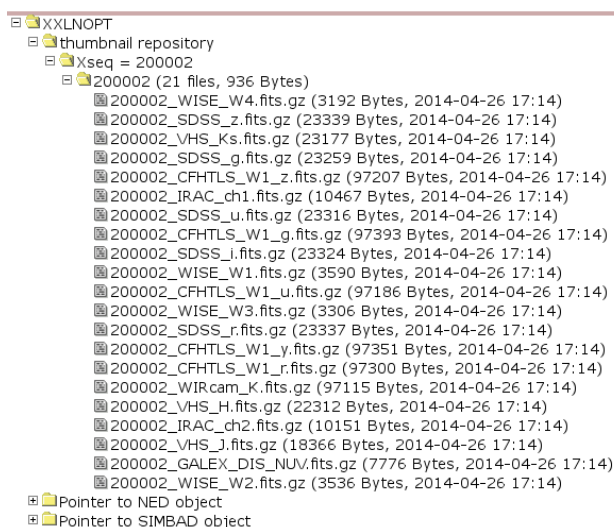
I provide here a very quick reference of the history of the visualization tool and the motivations which lead to a change. Since so far I have documented this in web pages and occasionally in internal reports, I will mainly refer to such external documents, which are listed in Table 2.

Originally, back in the time of XMDS, I started with the idea of using `ds9` to display existing thumbnail images, and overlaying onto them region files generated by database queries. This is viable using XPA but requires a lot of clicks.

Therefore I passed to experiment with *Java applets* which were used both for XMDS and XMM-LSS. An applet should be immediately usable by anybody inside a web browser. In practice this required some security customization at user end (java policy) or at the server end

Year	item	URL or reference
...-2006	ds9 regions from database	<a href="http://sax.iasf-milano.inaf.it/~lucio/LSS/LSS-Porq-Apr06/mostro.html">http://sax.iasf-milano.inaf.it/~lucio/LSS/LSS-Porq-Apr06/mostro.html</a>
2008-...	applet tests main page	<a href="http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/">http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/</a>
2008-...	user guide (v 24 onwards)	<a href="http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/#24more">http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/#24more</a>
2008-...	programmer doc (applet)	<a href="http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/progdoc.html">http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/progdoc.html</a>
2008	usage with XMDS	Report IV (Chiappetti , 2008a)
2008	early usage with XMM-LSS	Report V (Chiappetti , 2008b)
2010	see also (combo26/26b)	Report VII (Chiappetti , 2010)
2011	see also (combo26d/dx)	Report IX (Chiappetti , 2011)
2011	latest (XMM-LSS) applet	<a href="http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/combo26dx.html">http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/combo26dx.html</a>
2013	user & programmer doc (JWS)	<a href="http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/jwsdoc.html">http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/jwsdoc.html</a>
2013	latest (XMM-LSS) JWS	<a href="http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/combo26jws.html">http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/combo26jws.html</a>
2014	doc for this version	<a href="http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/new.html">http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/new.html</a>

**Table 2.** Reports and web pages about previous versions of the visualization tool (see also references therein)



**Fig. 1.** Sample tree for individual data products retrieval



**Fig. 2.** Sample tree for retrieval of a collective tar.gz with all data products

(signed applets) which is not always easy or immediately compatible.

A further security issue due to our choices for the Milan database, is that the applet won't talk directly to the database (avoiding individual login), instead a web form (where one enters the catalogue and X-ray source number) talks to a servlet in Milan, and the servlet talks to the database (so all mysql communication occurs locally on a single user). Applet and servlet talk over a socket. For what concerns images, their URLs are returned to the applet, which then fetches the images via http. The region information instead travels on the socket.

A next step was (rather recently) a migration to *JWS (Java Web Start)*. This should have avoided one problem with applets, i.e. if something gets stuck, one cannot reload the applet unless one terminates and restarts the browser. JWS applications instead run in a separate window (but are started from a web page, and the rest of our own protocol was unchanged). Unfortunately it looks like the capability of a given browser to use JWS is rather site dependent and requires customization at the user end.

So finally I decided to distribute the present *standalone application*, whose inner working is described in section 4, but is anyhow independent of browsers and should have no specific security requirements.

## 2. Quick start guide

The supplied application should work on any system having a Java runtime environment and presumably not having unusual firewall requirement. I have been using it under Java SE Runtime Environment (build 1.6.0\_10-b33).

You first retrieve the `.jar` file from the site listed in 2.1, and then start it as described in 2.3, which contains an example run. Only in case of difficulty you may want to test your environment as described in 2.2.

### 2.1. Retrieving the material

The application is entirely contained in a Java `.jar` file. You should be able to download it just clicking on this URL in this PDF file: <http://sax.iasf-milano.inaf.it/~lucio/temp/Java/Alone/alone27.jar>.

If you undergo difficulties go to the page <http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/new.html#quick>, and click on the link labelled "alone27.jar" in the second line.

Note that if you haven't already stored the access credentials in your browser, the page may require a consortium username (XXL) and password (xNewton).

Now jump to the quick start instructions in 2.3.

## 2.2. Testing

Only in case you experienced difficulties with the startup, you might want to run the following test.

- retrieve the following java source file <http://sax.iasf-milano.inaf.it/~lucio/temp/Java/Alone/prova.java>. (or click on "prova.java" on the third line of page <http://sax.iasf-milano.inaf.it/~lucio/temp/Java/TestWeb/new.html#quick>).
- compile it with `javac prova.java`
- run it with `java prova`

You should see in your terminal an output like this

```
DEBUG socket created
DEBUG socket bound to null
DEBUG got local address 0.0.0.0/0.0.0.0:60747
DEBUG socket connected
DEBUG input stream OK
DEBUG output stream OK
```

```
written to output channel message ABCD
written to output channel message XYZT then...
  read ABCD
written to output channel message ababab
written to output channel message zzzzxx
written to output channel message maimai
then I will read THREE from input and exit
  read XYZT
  read ababab
  read zzzzxx
```

The lines labelled "DEBUG" should be the same which you see when running the actual tool as explained in 2.3. If the actual tool did not display all 6 DEBUG lines, also the test program will stop before.

The test program is a silly self-contained java application, which has no relation whatsoever with the XXL database, the XXL data product web server, or the outside network. It just tries to create a socket, bind and connect it to the local address, then short-circuit an input and output stream on the socket.

If this is successful, it writes some text strings (first two, then other three) to the output stream and reads first one, and then other three from the input stream (the last one is never read). This emulates the way the actual tool works, which in turn emulates *internally* what the former (applet and JWS) tools did connecting with a servlet in Milan (the current application does not talk directly to any servlet in Milan, just uses http!).

If there is a failure on some of the first 6 lines, you should find the help of some system manager or experienced java programmer at your site, and try to tweak your system protections in such a way that the test program works. *I cannot tell you what to do, because here it works!* but I'm pretty sure that, if the test program works, also the actual program will do!



**Fig. 3.** Screenshot of the tool at startup, with a ramp in the main display pane

## 2.3. Using the tool

Once you have retrieved the .jar file, you run it as

```
java -jar alone27.jar &
```

This shall display some more or less copious console messages, which you usually ignore, and open a window like the one shown in Fig. 3. If you experience an early failure (the window does not appear, and/or there are less than 6 DEBUG messages in your terminal), go back to section 2.2 and try running the test program.

If everything is fine, you can proceed to a demo run as described here.

You will notice that the window is divided vertically in three areas (see also 3.1).

- on top there is a tab pane, whose content varies according to the selected tab. Initially it is set to the main display and contains a red-scale ramp image (a ramp image may be loaded when no FITS files are or can be accessed)
- in the middle there is a message area (with an optional scroll bar)
- on the bottom there is a region table, initially empty

The next step of the demo is to click on the third tab labelled "control panel". The content of the control pane at initialization is shown in Fig. 4.

The control pane appears divided vertically in 4 areas:

- on top there is a control area with two input areas, a Catalog menu and an Xseq input area, and 6 buttons

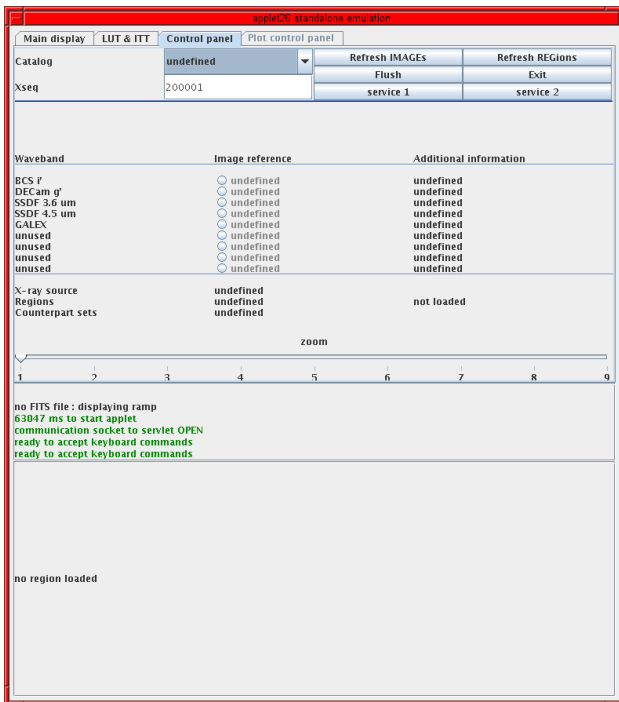


Fig. 4. Screenshot of the control pane at startup

arranged in 2 columns of 3 rows. Buttons not described in this section are explained in 3.1.3.

- just below it there is a list of images per band, which is described briefly for the demo and in detail in 3.1.4.
- then there is a short message area about regions etc.
- and on the very bottom there is a *zoom slider* which can be used to set the zoom of the image in the main display

In the next step of the demo you pull down the catalogue menu and select a valid catalog. Then you enter an existing *Xseq* in the input area. For instance do *TESTXOS* and *200015*. Note that if you put the mouse over any button, menu, etc. you should see a tool tip.

Valid catalogues are *XXLNOPT* and *TESTXON* in the North (“Lucio” or “Sotiria” style respectively) or *XXLSOPT* and *TESTXOS* in the South. Catalogues in the same area use the same *Xseq* (those in *XXLN* or *XXLS*). If you refer to a *seq* which exists in *north33* or *south33*, but not in *XXLN* or *XXLS* (because it has been removed by the overlap removal procedure) you might get later error messages when trying to retrieve images or regions.

The next step in the demo, once selected a catalogue and source, is to press the “Refresh images” button. This will load a list of images on the screen, and might possibly also display one in the main display pane. Fig. 5 shows how the control pane looks at this stage.

The first column lists the catalogues associated with the chosen X-ray catalogue. The second column (image reference) lists the existing files. Note that a radio button in front of *one* image may be ticked. If this corresponds

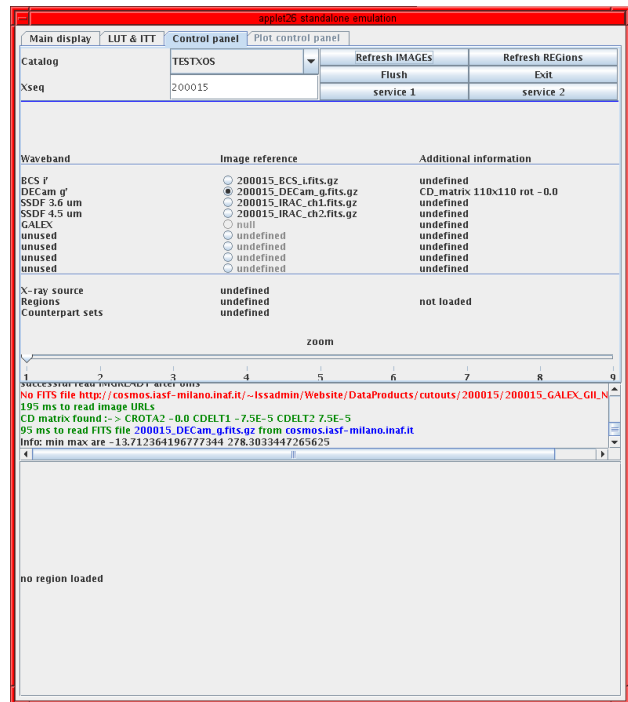


Fig. 5. Screenshot of the control pane after doing a “Refresh images”

to an existing image, the third column will contain some WCS summary. A “null” entry means there is no image at all for that catalog/band for that particular source. You can load another image clicking on its radio button. The position will be remembered (i.e. if you select the third image, and then do a refresh for another source, it will attempt to display the third image even if not existing, so you’ll have to click on another image manually).

Note that (more details in 3.3.1) whenever there are more bands per catalogue, only one is selected according to a pre-configured priority (so for instance if there is an *i*’ image that will be selected, but if missing maybe the *g* band is selected). *The only way to look at the other bands is to download home the data products from the database (as explained in 1.1).*

If you want you might have a look at the main display now, but for the demo I suggest you first click also on the “Refresh regions” button. This will load data in the bottom region table, as shown in Fig. 6. The region table will remain always visible irrespective of the tab pane you display.

The content of the region table will be described in detail in 3.1.7 and only en passant during the demo.

At this point, before going to the main display, you should click on the second tab labelled “LUT & ITT”. Fig. 7 shows the more or less self-explanatory content of this pane, which is explained in detail in 3.1.2. You are free to play around, but I suggest the first time you limit yourself to ticking the “log” scaling in the right hand column,



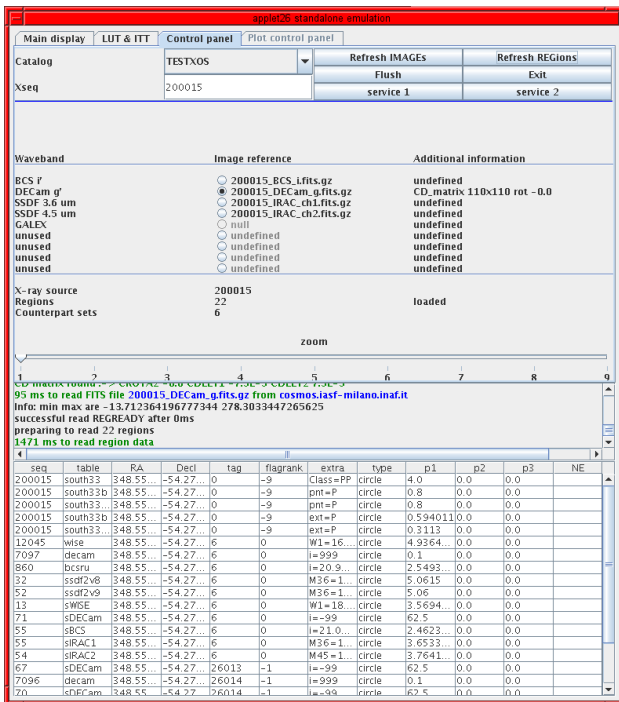


Fig. 6. Screenshot of the control pane after doing a "Refresh regions"

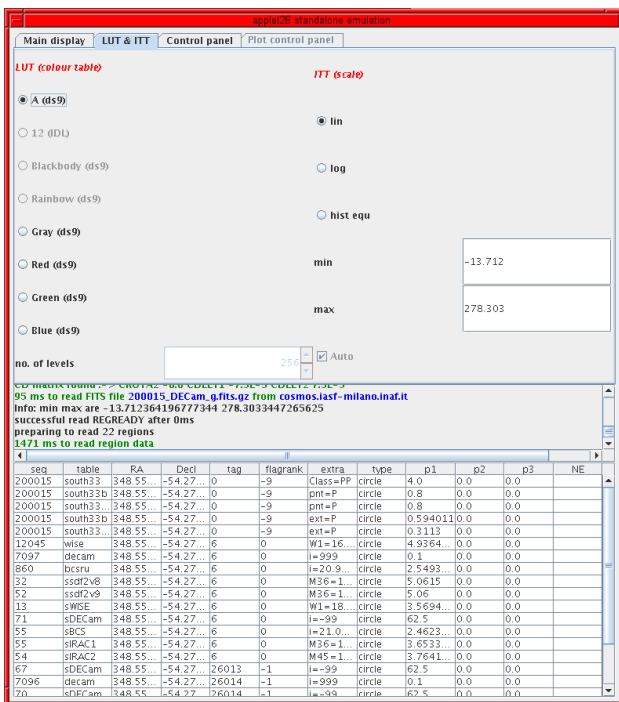


Fig. 7. Screenshot of the LUT & ITT pane

After each "refresh images" the min and max are updated to the values of the loaded image. If you tick on "log scaling" the minimum is reset to a safe, not negative and non zero value.

At this time you can look at the main pane, which will have a look as shown in Fig. 8, with a tiny image in a corner with some coloured regions superimposed.

At this stage you would want to zoom the image. There are several ways of doing it. Ideally one uses keyboard accelerators, but if some reason they won't work one has to resort to the zoom slider in the control panel. A zoomed image is shown in Fig. 9.

- The standard accelerators to zoom in and out are the + (plus) or - (minus) keys, with the mouse positioned anywhere inside the java window.
- One can also type a digit from 1 to 9 to zoom directly to a given (linear) zoom factor.
- Otherwise one goes to the control pane and drags or clicks on the zoom slider.
- +/- accelerators are the only way to zoom to/from a zoom factor above 9
- Zooming to a large factor will enable scroll bars in the main display pane
- Choice of an optimal zoom depends on the size of the original thumbnails in pixels.
- There are other (non-zoom) keyboard accelerator commands described later in 3.2.2.

Images are displayed oriented in XY pixels. Clicking on a position with mouse button 1 (MB1) will display the coordinates (also RA and Dec exploiting the image WCS) in the message area. Clicking with MB2 will temporarily hide the regions. The orientation disappears and the regions reappear moving the mouse.

The orientation marker is one red line going north and one green line going east, both 5" long (which allows to appreciate the scale of a zoomed image).

Note that the current tool supports only floating point (BITPIX=-32) images, and the RA--TAN DEC--TAN WCS (gnomonic projection). Some catalogues have thumbnails not consistent with these requirements (VHS and VIKING are not float, UKIDSS uses the ZPN WCS, and WISE the SIN WCS), and are therefore flagged as "unsupported". They will give error (red) messages in the message area, might or might not display, and if displayed will not allow WCS coordinate conversion (therefore region overlays are not allowed).

If you now look at the typical display, e.g. in Fig. 9, you might be curious to know what are the various coloured regions, and how to interact with them.

First of all the colour of a region corresponds to a catalogue, according to a configuration described in 3.3.2. The shape and size also depends on catalogues. Error boxes or ellipses are currently not supported (intended for radio sources). Fixed square markers are intended for SIMBAD or NED positions. Circles may be position uncertainties in arcsec (e.g. for X-ray sources) or be function of magnitude (larger for brighter sources). For the time being be contented to know that:

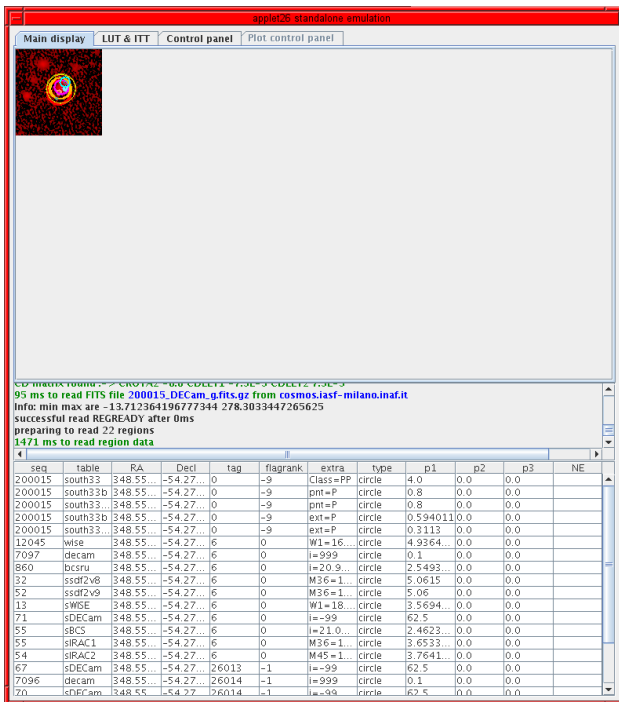


Fig. 8. Screenshot of the main display with image (and regions) loaded at no zoom

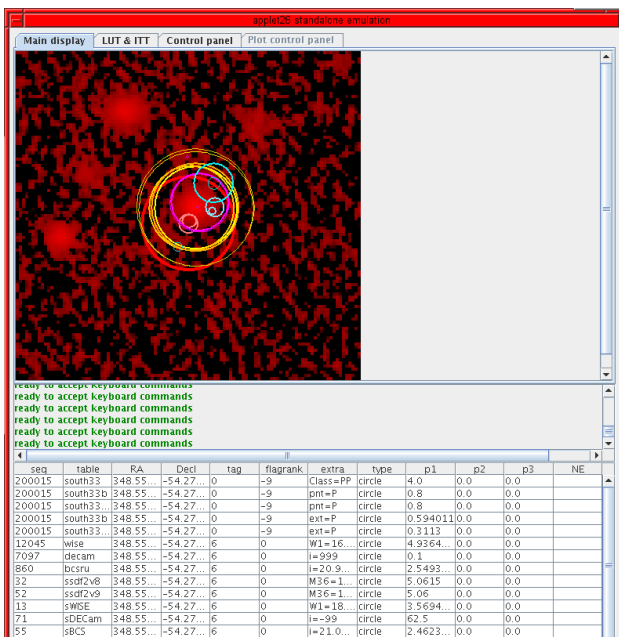


Fig. 9. Screenshot of the main display with a zoomed image

- thick red circles correspond to the X-ray ”official” position of  $X_{seq}$ . The size is the largest among the inter-band (soft-hard)  $maxdist$  and  $4''$
- thin red circles (only for XXLNOPT or XXLSOPT) are for neighbouring X-ray sources including those removed because of pointing overlaps. Size as above.

- pale pink and azure circles are for the individual soft and hard, pointlike and extended position (size is either the nominal position error  $radec_e$  or the extent  $corerad$ )
- other coloured circles are for non-X-ray catalogues (with radius function of magnitude)

To know which region is which object you may either:

- click near (*rather near*) the centre of a region (which is difficult if many overlap closely). The clicked region remains highlighted in thick colour, and the other are temporarily dimmed. The selected source is highlighted in the region table
- click in the region table in the first column labelled ”seq”. That will highlight an individual source
- click in the region table in the second column labelled ”table”. That will highlight all the sources coming from the indicated database table. This is useful to get empirically the colour-table association
- click in the region table in the fifth column labelled ”tag”. You will notice that more entries share the same tag. The tag is actually the  $seq$  in the GCT underlying the multiwave catalogue. There is one record per *counterpart set* with a counterpart in each specific table.
- Note that there is a primary highlight on the clicked table row, and a secondary highlight (the word FOUND in the last column) for all other selected entries
- Note also that, if an entry is selected, clicking on another column of the same row has no effect (click on another row and than back in the wished column; this is java behaviour)

If you got so far, everything should be functioning OK, and you should have gained enough practice to use the tool. To know more read next section.

Ah yes ... I forgot. To exit click the ”Exit” button in the control pane (or just close the java window).

And you might be interested to know you can run two independent versions of the jar in two separate windows side by side. This can be useful e.g. to compare ”parallel” catalogues like XXLNOPT and TESTXON on the same source, to see which new counterparts there are, or how they are differently ranked.

### 3. Command reference

#### 3.1. Layout

As anticipated in 2.3 the java window is divided vertically in three areas (see Fig. 3), a region table on the bottom (see 3.1.7), a message area in the middle (see 3.1.6) and a tab pane on top. The tabbed pane can cycle among a main display (see 3.1.1), a LUT & ITT pane (see 3.1.2) and a control pane, which in turn contains an input area with buttons (see 3.1.3), an image list (see 3.1.4) and other info (see 3.1.5).

### 3.1.1. Main display

The main display is where the current thumbnail image is loaded (see e.g. Fig. 8 or 9). If no FITS image can be loaded it displays a ramp image. If nobody appears here you have done some mistake (which perhaps can be cured with repeated clicks on the "Flush" button, or perhaps just by exiting and re-entering).

The image can be shown at various zoom factors, and if necessary vertical and horizontal scroll bars will appear. The image can be displayed with various colour tables (LUTs) and with various scalings (ITTs) as controlled in the LUT & ITT pane (see 3.1.2).

Regions can be overlaid on the image and temporarily be hidden. The image is always shown oriented in XY pixels, but one can obtain a North-East marker, which also works as a scale marker with MB2 (see 3.2.3).

One can change the image and region from the control pane (see 3.1.3), via buttons (see 3.2.1), mouse (see 3.2.3) or keyboard (see 3.2.2).

### 3.1.2. LUT & ITT

The LUT & ITT pane (see Fig. 7) should be self-explanatory and provides a subset of functions usual in display tools like ds9. The left hand side concerns colour tables, and the right hand side the scaling.

The main display is initialized with the red colour table (despite what the radio buttons in the left column say). After that one can change the LUT ticking the appropriate radio button (if active). Currently all LUTs have 256 levels. LUTs relate an intensity number between 0 and 255 to an RGB colour. Supported LUTs mimic either ds9 or IDL LUTs.

The scaling maps the actual content of an image to the intensity number. Linear, logarithmic and histogram-equalized scaling is supported, between the indicated minimum and maximum. Min and max are set when reading a new image. They can be changed by the user. To reset them to the image min and max just reload the image. Choosing log scaling will set the minimum to a sensible non-negative value.

### 3.1.3. Control pane buttons

The top of the control pane contains a menu, an input area and 6 buttons (see Fig. 4). This is the main interaction area, although some functions are possibly also or only via mouse (see 3.2.3) or keyboard (see 3.2.2).

The choice of the working catalogue from the menu is mandatory. Please select one of the indicated catalogues and do not leave it to the initial "undefined" value.

The choice of the X-ray source `Xseq` is also required and possible only here.

The function of the 6 buttons (of which one usually uses only the self-explanatory "Refresh Images", "Refresh Regions" and "Exit") is explained in 3.2.1.

### 3.1.4. Image list

The image list area (see Fig. 5) contains 3 columns labelled "Waveband" (improperly), "Image reference" and "Additional information". Currently the image list area supports up to 14 rows.

- The first column is loaded at "Refresh images", whenever a valid catalogue is chosen, with the list of potential non-Xray catalogues. The description may say which is the *preferred band* or may contain a notice that a given image is *unsupported*. As explained in 2.3, images which are not floating point, or do not have gnomonic projection RA--TAN DEC-TAN WCS are currently not supported. They may or may not display, but even if they display conversion from pixel to sky position (and hence region overlay) will not be possible.
- The second column contains a FITS file name and a radio button, and is loaded at "Refresh images". The program will look for an image for the given X-ray catalogue and a "preferred band", then, if not found, it will fall back to the next band in a configured preference order (see 3.3.1). Only if there is no image at all it will display a "null" entry and the radio button will be disabled. The message area will provide red error messages for all images which are looked for and not found. One radio button at a predefined position will be ticked (usually the second one), and the program will attempt to load the image in the main display. If there is a null image at such position, you should tick the radio button of an existing image (and such position will be later remembered). Note that the URL (filenames) of all images in the preferred band which exist for one X-ray source are loaded once. One can then display each of them ticking on the appropriate radio button (or using keyboard accelerators, see 3.2.2), without the need to repeat "Refresh images".
- The third column is loaded when a given image is loaded in the main display (and then remembered). It will contain a summary about the image size in pixel and the way its WCS keyword implement rotation from XY to RA,Dec.

### 3.1.5. Rest of control pane

The bottom part of the control pane (see Fig. 4 or 5) contains a small status area about the regions, and a *zoom slider*.

The status area contains `Xseq`, the number of regions i.e. objects (more or less, possible counterparts) present,



and the number of counterpart set. A counterpart set groups the objects which are the same and associated across different bands. For counterparts vs counterpart sets see 3.1.7.

### 3.1.6. Message area

The message area contains all messages generated by the program. A scroll bar will appear when needed to add more messages, and will be usually positioned at the bottom (last message).

The message area can be cleared clicking mouse button 3 (MB3) in the main display, using keyboard accelerator **z** or pressing the "service 1" button in the control pane (each of these has also some side effects).

Messages are of three kinds:

- Normal informative messages are in black and usually return some value requested by some action
- Information messages are in green and correspond to the successful execution of some action
- Error messages are in red. Most errors are physiological (e.g. no image found where expected, or clicking on a position where there is no source) and are corrected either automatically (e.g. skipping to the next band image) or just continuing.

### 3.1.7. Region table

The region table (see Fig. 6) is loaded whenever a "Refresh regions" is done. It is a standard Java SWING table with 12 columns and as many rows as necessary. Eventually a vertical scroll bar will allow to see the "regions" which do not fit on the screen. Using the mouse on the table header you can enlarge or shrink a column if you need to read long strings contained in such column or an adjacent one.

There is one row per region displayed on the main display. The *correspondence between regions and counterparts or counterpart sets* in the actual catalogues has to be explained.

- Each catalogue has an underlying GCT (Generalized Correlation Table); actually **TESTXON** and **TESTXOS** do not exist as catalogues but currently only as GCTs
- Each GCT contains one or more records per X-ray source. These records are termed "counterpart sets"
- All counterpart sets pertaining to the same X-ray source share the same **Xseq**, which is actually the **seq** in the primary member table (aka PBDA), i.e. **north33** or **south33**. The internal GCT **seq** is instead different for each counterpart set.
- Each counterpart set has a number of member tables, and contains for each member table the **seq** pointing to the member table record (all scientifically interesting values are got from the latter table).
- The first member (PBDA) pointer is the same for all counterpart sets pertaining to the same **Xseq**, actually

it *is* such **Xseq**. Therefore this information is displayed *only once* in the region table, in the very first row.

- The next two member tables are the soft and hard X-ray tables (e.g. **north33b** and **north33cd**). These pointers are also the same for all counterpart sets pertaining to the same **Xseq**. For the purpose of the region table, they appear on top next to the primary entry and with the same "tag" set to zero, but are duplicated for each band. *There is one entry for the XAMIN point-like fit, and one entry for the extended fit.* The "extra" column is labelled either **pnt** or **ext** and then contains the actual PE classification used in the database. This way one can appreciate the position of the X-ray source in both bands and in both the chosen and discarded classification.
- The next member table is a "duplication" of the main X-ray table (e.g. **north33dup**). This information is usually also the same for all counterpart sets pertaining to the same **Xseq**, but there may be additional dummy records. This information is present only for "my" catalogues **XXLNOPT** and **XXLSOPT**, but not for the ones I faked using SF's tables. The "dup" entries points to the **seq** of the X-ray sources removed by the pointing overlap removal. If the chosen pointing overlaps just with another one, there is one "dup" entry tagged with "tag" zero. If it overlaps other pointings, these entries have dummy records with a different "tag" (i.e. **GCT.seq**).
- So each counterpart set has some X-ray members (the merged X-ray table, the soft and hard X-ray table, and eventually the "dup" pointer), and a number of non-X-ray members. It has also its own **GCT.seq**.
- For all non-X-ray members there is a row in the region table. The "tag" column contains the same non-zero **GCT.seq** for counterparts belonging to the same counterpart set.
- On the contrary, rows pertaining to X-ray source appear only once (or twice for P/E cases) and are usually tagged with tag=0 and appear on top.

Said that, one can explain the purpose of the various columns:

- Column 1 **seq** is the object **seq** in the member table,
- and column 2 **table** is the name of the member table
- Column 3-4 **RA** and **Decl** are the coordinates of the object (centre of the region)
- Column 5 **tag** is the **GCT.seq** of the counterpart set, or zero for entries common to all counterpart set (X-ray members) as explained above
- Column 6 **flagrank** is the rank column if present in the GCT. It is artificially set to -9 for X-ray member entries common to all counterpart sets (those tagged zero). Note that for **XXLNOPT** and **XXLSOPT** I computed no ranking, so it is artificially set to 66 for all entries. The rank has been computed "my way" for

TESTXON and TESTXOS, as detailed in Report XIV (Chiappetti, 2014).

- Column 7 **extra** provides additional information in the form of a string. Usually I report here one magnitude, or one magnitude and the id, or an identifier, or the P/E classification.
- Column 8 **type** is the shape of the region (see 3.3.2)
- Column 9-11 **p1** to **p3** are the size parameters. For circles it is the radius rescaled to arcsec, for points (drawn as filled squares) should be in unzoomed screen pixels.
- Column 12 **NE** is usually empty. It is dynamically set to FOUND for the records which are explicitly or implicitly selected.

Selection in the region table occurs according to the following rules:

- Clicking on the **table** column will primarily select the row where one clicks, but secondarily select (setting last column to FOUND) all objects coming from the selected table
- Clicking on the **tag** column will primarily select the row where one clicks, but secondarily select (setting last column to FOUND) all objects belonging to such counterpart set.
- Clicking on any other column will select just the object where one clicks
- Note that clicking on a different column of a row already selected will have no effect (this is a java artifact: click on any other row, and then on the wished column in the old row)
- One can also select a source clicking near its centre in the image display. Note that it could be difficult to discriminate objects whose position are rather close.

### 3.1.8. Highlighting modes

Highlighting of selected sources in the image display occurs in two modes. The modes are toggled using the **g** keyboard accelerator.

In the default dimmed mode the selected region(s) remain in their normal colour, with eventually a larger thickness. All other regions are temporarily grayed out.

In primary highlight mode unselected regions remain in their natural colour, while the selected ones appear in thicker white or yellow, respectively for primary or secondary selections.

Note that selecting regions from the image may be difficult when they overlap, however, since the selection works if one ticks within 2 image pixels from a region centre, one can zoom the image for an easier selection. It is possible that clicking on a position selects more regions, in this case the closest one to the clicked position will be the primary selection, and the other secondary selections.

## 3.2. Functions

One can interact with the program via buttons, keyboard accelerators or mouse clicks. We give the various functions by modality and then in Table 3 we summarize them by function in a more or less practical order.

### 3.2.1. Function listing by button

There are 6 buttons in the control pane, of which usually only the first 3 are used:

- Refresh Images retrieves the URLs of all existing images for the given **Xseq** and lists them in the image list area underneath. In addition it tries to load in the main display the ticked image. If that corresponds to a non-existing image (null URL), please manually cycle to a catalogue with an existing image. For keyboard accelerators for this function see 3.2.2
- Refresh Regions retrieves the regions for the given **Xseq** and catalogue, loads them in the region table and overlays them on the main display. **r** is the keyboard accelerator for this function.
- Exit terminates the application. There is currently no keyboard accelerator for this function, while one can exit just closing or killing the java window with window-manager-dependent ways (the application is set to "exit on close").
- Flush: this command (and its keyboard accelerator **f**) are normally not used. They flush the internal socket stream used to emulate the original applet/servlet communication. Because of the emulation this is unlikely to get cluttered, unless of some misconfiguration.
- service 1: this command is normally not used. It performs two functions. It clears the message area (see 3.1.6) and toggles an internal debug flag. So clicking it once will most likely increase verbosity of the messages, and clicking it again restore the original verbosity.
- service 2: this command is currently useless. It will display a message about the socket status, but since the socket streams are now merely internal, this is of no interest at all.

### 3.2.2. Function listing by keyboard accelerator

Keyboard accelerators are active whenever the main display pane is active. They are called typing one character on the keyboard, with the mouse pointed anywhere in the java window (not only in the main display, but also in the message area and region table). Keyboard commands are listed in alphabetic order.

- **f** (mnemonic for "flush") flushes the internal socket streams (same as "Flush" button, see 3.2.1)
- **g** will toggle the *highlighting mode* between "highlight" and "dimmed" modes (see 3.1.8)
- **h** (mnemonic for "headers") will load all image URLs, i.e. perform a "Refresh images"

Function	Button/Menu	Keyboard	Mouse
setting catalogue	menu in control pane		
setting <b>Xseq</b>	input area in control pane		
load image list	"Refresh images" button	<b>h</b> accelerator	
advance waveband pointer	click radio button in image list	<b>k</b> accelerator	
decrease waveband pointer	click radio button in image list	<b>j</b> accelerator	
load current image	click radio button in image list	<b>i</b> accelerator	
load region table	"Refresh regions" button	<b>r</b> accelerator	
re-center regions		<b>s</b> accelerator	
set colour table	radio button in LUT & ITT pane		
set scaling	radio button in LUT & ITT pane		
edit min/max	input area in LUT & ITT pane		
set zoom		<b>1 to 9</b> accelerator	drag zoom slider
increment zoom		<b>+</b> accelerator	click zoom slider right
decrement zoom		<b>-</b> accelerator	click zoom slider left
image position readout			MB1 in display area
show image orientation			MB2 in display area
hide regions	"service 1" button		MB3 in display area
toggle highlight mode		<b>g</b> accelerator	
direct region highlight			click MB1 2 pixels from region centre
inverse region highlight			click any column in region table
counterpart set highlight			click "tag" column in region table
database table highlight			click "table" column in region table
clear message area	"service 1" button	<b>z</b> accelerator	MB3 in display area
exit program	"Exit" button		
flush socket	"Flush" button	<b>f</b> accelerator	
toggle debug mode	"service 1" button		

**Table 3.** Summary of commands by function

- **i** (mnemonic for "image") will load in the main display the image for the currently selected catalogue/ survey/ band (incorrectly termed collectively as "waveband"). Useful after one has "advanced" the waveband using the **j** or **k** command
  - **j** will decrease the waveband pointer and
  - **k** will increase the waveband pointer, i.e. they act if the previous or next radio button in the image list has been ticked. They allow to cycle thru the various images without leaving the main display pane
  - **r** (mnemonic for "regions") will load the regions, i.e. perform a "Refresh regions"
  - **s** is an alternate way of redisplaying regions already loaded, if by any reason they disappear or are misplaced. It will redisplay them centered in the display area (typically after a zoom change).
  - **z** currently clears the message area
  - **1 to 9** set the zoom factor of the display area directly to the given (linear) zoom level
  - **+** increases the zoom factor by one (even beyond 9)
  - **-** decreases the zoom factor by one
- 3.2.3. Function listing by mouse button
- Mouse clicks are effective in the image display area, in the region table, and of course to interact with the various widgets (buttons or radio buttons) in the various panes, inclusive of the zoom slider. I list here only the non self-explanatory functions.
- Mouse button 1 (MB1) in the image display will always read the pixel and sky coordinates (and the intensity number value) at the clicked position, and write it in the message area. Moreover if the position is close to a region centre, it will highlight the object (in the region table and on the main display)
  - Mouse button 2 (MB2) in the image display will always read the pixel and sky coordinates as MB1, but also display an orientation marker consisting of one red line going north and one green line going east, both 5'' long (which allows to appreciate the scale of a zoomed image). The marker disappears at the next click, or moving the mouse out and in the active display
  - Mouse button 3 (MB3) in the image display temporarily hides the region overlay and at the same time clears the message area. Regions reappear at the next click, or moving the mouse out and in the active display
  - clicking the mouse (MB1) in the table column of the region table will select and highlight all sources in the given member table
  - clicking the mouse (MB1) in the tag column of the region table will select and highlight all sources in the given counterpart set

- clicking the mouse (MB1) anywhere else in the region table will select and highlight the current source, as described in 3.1.7 and 3.1.8
- dragging the mouse (any button) in the zoom slider will set the zoom factor to the closest value
- clicking the mouse (any button) in the zoom slider right or left of the current position will increase or decrease the zoom factor by one. Note that the zoom slider initially goes from 1 to 10, but if the zoom factor is increased beyond it with the + accelerator it will show a larger range (reset to 1-10 as soon as it drops below 10)

### 3.3. Conventions and configuration

#### 3.3.1. images per catalogue

As explained in 1.1 and Table 1, there are several catalogues, with sub-surveys and bands for which SF collected cutouts. While one can retrieve all of them via the database, it is simpler if the visualization tool allows access to just one image per case, selecting the bands or surveys in a preference order. So *only* the first image encountered in the preference order is displayed.

The preferences are taken from configuration files described in 4.1.2 and currently are those listed in Table 4. Catalogues are listed in the order shown in the image list (optical, NIR, IR then UV). Note that the ABC fields are handled together (only one best image is shown for all of them where overlapping). Note also sometimes the priority order is survey then band, and sometimes is band then survey or field.

#### 3.3.2. colour per table

The choice of the colour and thickness of the regions associated to a given *database* member table are controlled by the configuration file described in 4.1.1. They are described with the other region characteristics in the next section.

In general the colours chosen are such that they are equal for database tables of equivalent waveband coverage in north or south, and are similar with different shades (marked with superscripts 1,2,... in Table 5) if there are more tables for the same waveband. Also the primary X-ray source is thicker, the tables (not publicly released) supplied by SF have a medium thickness and a brighter colour, the original (but publicly released) database tables are thin and darker.

#### 3.3.3. regions per table

The choice of the shape and size of the regions associated to a given *database* member table are controlled by the configuration file described in 4.1.3. They are described

Catalogue	preferred bands
<b>North</b>	
CFHTLS_D1	i y g r z u
CFHTLS_W1	i y g r z u
CFHT A/B/C	WA_g WB_g WC_g WA_r WB_r WA_z WB_z
SDSS	i g r z u
UDS	i z R V B
UKIDSS	DXS_K UDS_K DXS_J UDS_J
VHS	$K_s$ H J
VIDEO	K H J Y z
VIKING	$K_s$ H J Y Z
WIRcam	K
IRAC 3.6 $\mu$ m	ch1
IRAC 4.5 $\mu$ m	ch2
GALEX	DIS(NUV FUV) AIS(NUV FUV) GII MIS
WISE	W1 W2 W3 W4
<b>South</b>	
BCS	i g r z
DECam	i g r z
VHS	$K_s$ H J
SSDF 3.6 $\mu$ m	ch1
SSDF 4.5 $\mu$ m	ch2
GALEX	AIS(NUV FUV) GII(NUV FUV) MIS
WISE	W1 W2 W3 W4

**Table 4.** Preference order of bands/surveys per catalogue (as shown). GALEX sub-surveys GII and MIS have only NUV data.

here, namely in Table 5, as currently configured, together with the colour and thickness.

When the size is indicated as  $f(\text{mag})$ , note that for my tables it refers to *raw* magnitudes, while for SF's tables it refers to *homogenized* magnitudes, therefore sources from the equivalent tables may be shown with more or less different radii according to the effect of homogenization. The scaling by magnitude is given by  $f(\text{mag}) = (26 - \text{mag})/2$ , with a lower limit of 0.1". Indications like  $g(\text{flux})$  and alike indicate some provisional form of (logarithmic) scaling by flux (to be dismissed in the future when only homogenized magnitudes will be used).

The content of the actual configuration files is subject to changes without notices (not that I intend to make changes, but could be requested to do so) and without updates of the present document.

## 4. Technical reference

For details I refer to the source code, or to web documentation in the last URL in Table 2.

The standalone application has *not* been redesigned in an optimized way. In particular it does *not* talk directly with the database over JDBC, which would be possible, but would require an individual `mysql` authentication (either registering specific hosts, which is painful to maintain, or allowing access from any host, which is unsafe). Instead it has been adapted from the latest applet/JWS code with minimal changes.

Table	colour/thickness	shape	size
X-ray merged	red thick	circle	maxdist or 4''
X-ray dup	red thin	circle	maxdist or 4''
X-ray soft pnt	pink mid	circle	radec_e
X-ray hard pnt	azure mid	circle	radec_e
X-ray soft ext	pink mid	circle	corerad
X-ray hard ext	azure mid	circle	corerad
<b>North or both</b>			
w1t7	magenta <sup>1</sup>	circle	f(i)
CFHTLS_W1	magenta mid	circle	f(i) or f(y)
CFHTLS_D1	magenta	circle	f(i)
CFHT A B C	magenta-blue mid	circle	f(g)
sdssdr10	magenta-red <sup>1</sup>	circle	f(i)
SDSS	magenta-red mid	circle	f(i)
UDS	magenta-dark mid	circle	f(i)
wircam	orange-yellow <sup>1</sup>	circle	f(K <sub>s</sub> )
nWIRcam	orange-yellow mid	circle	f(K <sub>s</sub> )
ukidssdr10	yellow-green <sup>1</sup>	circle	f(K)
UKIDSS	yellow-green mid	circle	f(K)
VIDEO	green mid	circle	f(K)
swiredr6	yellow <sup>1</sup>	circle	g(flux)
ssdf2v9	yellow <sup>2</sup>	circle	f(mag <sub>36</sub> )
irac2v0	yellow <sup>3</sup>	circle	f(mag <sub>36</sub> )
ssdf2v8	yellow <sup>4</sup>	circle	f(mag <sub>36</sub> )
IRAC1	orange mid	circle	f(mag <sub>36</sub> )
IRAC2	orange <sup>1</sup> mid	circle	f(mag <sub>45</sub> )
wise	yellow <sup>5</sup>	circle	f(W1)
nWISE	yellow mid	circle	f(W1)
galexgr6	cyan <sup>1</sup>	circle	h(flux)
omsuss	cyan <sup>2</sup>	circle	k(flux)
GALEX	cyan mid	circle	f(NUV)
<b>South</b>			
bcsru	magenta <sup>1</sup>	circle	f(i)
bcs1mu	magenta <sup>2</sup>	circle	f(i)
sBCS	magenta mid	circle	f(i)
decam	magenta-red <sup>1</sup>	circle	f(i)
sDECam	magenta-red mid	circle	f(i)
sWISE	yellow mid	circle	f(W1)
sIRAC1	orange mid	circle	f(mag <sub>36</sub> )
sIRAC2	orange <sup>1</sup> mid	circle	f(mag <sub>45</sub> )
sGALEX	cyan mid	circle	f(NUV)
<b>External</b>			
ned	orange <sup>1</sup>	point	4 pix
simbad	orange <sup>2</sup> mid	point	3 pix
usno	orange-red mid	circle	f(R)
marseillespec	blue	point	2 pix

**Table 5.** Region characteristics per table. Where thickness (thick, mid, thin) is not indicated it defaults to thin. Superscripts indicates different shades of the quoted colour. "North or both" means that the listed database tables may cover only the northern area (either because the survey covers only such area, e.g. CFHTLS, or because SF supplied separate north and south files), or both (this is typical of my tables, where there is e.g. a single `galexgr6` table, while SF supplied separate `GALEX` and `sGALEX`; table names are as in Report XIV, Chiappetti 2014). These colours do not consider those used for highlighting as described in 3.1.8.

- One change has been to incorporate the choice of catalogue and `Xseq` in the control panel (avoiding the usage of a web form)
- Also the listing of possible image sources (catalogues) has been adapted to XXL, with partial hardcoding (the allowed X-ray catalogue names are hardcoded in the Java code, while the non-X-ray table characteristics are hardcoded in configuration files)
- The protocol used in the applet and JWS versions, by which the main application sent opcodes and retrieved results to a servlet has been maintained, but with an *internal emulation*. This can be a weak point since it requires an internal local socket (the one which is tested as described in 2.2) however it minimized re-coding.

So in a nutshell the current application works in the following way:

- Once an X-ray catalogue and `Xseq` have been chosen, the list and characteristics of associated thumbnail catalogues is retrieved *via http* by a *configuration file residing in Milan on my own server*.
- When doing a "Refresh images", the list of possibly available thumbnail images is tested and retrieved *via http* from *the Milan database server*.
- Since the Milan database server does *not* run Apache but Tomcat, it could not natively support http to static files. In fact when retrieving data products within the database (section 1.1), the database server accesses them as local files, which is optimal, allows generation of collective tar.gz files, and allows testing of existence of the files. However Tomcat has been ad-hoc configured to support static data products for the purpose of this tool.
- "tested and retrieved" means the following. For what concern testing, since the existence of the file pointed by an URL is not immediate, I devised a simple java method which does that trying to open an input stream on the fits.gz URL. The stream is then closed and no retrieval occurs at this stage.
- If it fails, the thumbnail is considered not existing. The list of existing thumbnail URLs for the given X-ray source is retrieved once and stored internally, and displayed in the image list on the control pane.
- An image is retrieved *via http* from the stored URL whenever its display is requested (this is identical to the previous applet and JWS versions)
- Instead when a "Refresh regions" is requested, the catalogue name and `Xseq` are passed *via http* to a CGI script running on *my server*.
- the CGI returns just the name of a *temporary text file* which resides on *my server* and is automatically deleted after one minute.
- The application then retrieves *via http* the content of the temporary text file from a predefined location on *my server*. This content is the number and value of the



- regions (in the same format which the applet and JWS used to read from a socket connected to a servlet).
- What the CGI *should* do is to talk to a "reduced servlet" which also runs on *my server*. This servlet (`servlet27`) is the only one which talks (locally) to the database `mysql` server.
  - The servlet gets the names of the GCT underlying a particular X-ray catalogue and of the member tables. This means `TESTXON` and `TESTXOS` are registered in the `VirtTables` administrative table although not advertised to user nor configured as `VIEWS`.
  - The servlet issues a query to the GCT to retrieve all the records for the counterpart sets associated to `Xseq`. Then for each counterpart and each member table, it issues a query to the member table with a specific syntax to retrieve the object position and the characteristics (e.g. error radius or magnitude) necessary to draw the regions.
  - The colour and thickness used for each member table are retrieved *directly* (by the standalone application) *via http* by a *configuration file residing in Milan*
  - Instead the member-table specific characteristics of the query need to be known to the servlet only (which returns pre-cooked text). They are considered *configuration items* as well, and are read by the servlet as `HttpServletRequest` parameters.
  - This is totally analogous to what done by the full-fledged servlet (`servlet26`) used by the applet and JWS. However, the latter can easily read the parameters from the web form which invokes the servlet (where they are stored as hidden controls).
  - The problem is that here `servlet27` is not invoked by a form, but by the CGI.
  - The CGI therefore first generates a temporary HTML file which contains the hidden controls, as well as catalogue name and `Xseq`.
  - The CGI uses a PERL script `formfind` (retrieved from the network) to parse the HTML file and extract all parameters. A custom awk script reformats the list of parameters as `-d` arguments to `curl`. Finally the CGI uses `curl` to invoke the servlet with the full list of parameters (they are currently 380!), and saves the output to the temporary text file.

#### 4.1. Configuration items

Configuration items are currently maintained centrally in Milan. Such items are the following and kept in the following configuration files.

##### 4.1.1. Member colour configuration

The member table colour configuration file, read via `http`, is specific of `XXL` (i.e. different from the one used for `XMM-LSS`). Its name is hardcoded in java code (`xxltable.config`). It contains one line for each possible

member table (in any catalogue) in the following format, which gives the thickness and the RGB colour of regions associated to each table :

```
tablename thickness red green blue
e.g. north33 4 1 0 0
```

The combined result of the current colour and shape configuration is shown in Table 5.

##### 4.1.2. Catalogue configuration

The thumbnail catalogue configuration file, also read via `http`, is specific of this standalone application and allows to customize the possible non-X-ray catalogues for which there is a thumbnail available for a possible catalogue. There is one configuration file per catalogue, with a name like `catalogue.config`. The files for `TESTXON` and `TESTXOS` are soft-links to the ones of `XXLNOPT` and `XXLSOPT`, respectively. The files lists the possible thumbnails allowed, and indicate the priority of the bands. These information are used to build the thumbnail image names indicated in 1.1, and displayed in the control pane. There is one line per thumbnail catalogue, with the following colon-separated format, where instead *bandlist* is a comma-separated list :

```
label:catalogues:bandlist e.g.
CFHTLS W1 (preferred i'):CFHTLS.W1:i,y,g,r,z,u
or GALEX:GALEX:DIS_NUV,DIS_FUV,AIS_NUV,...
```

Note that the survey specification is coalesced with the band specification (e.g. in the `GALEX` example). The first line means that if an *i*-band thumbnail exists for `CFHTLS W1`, the java tool will show it *only*, otherwise it will try the *y* band, then *g* etc. The second line means that for `GALEX` it will prefer `NUV` data from the `DIS` survey, then `FUV` from `DIS`, then `NUV` from `AIS` etc.

The result of the current preference configuration is shown in Table 4.

##### 4.1.3. Member table configuration

This configuration is concerned with region shapes and sizes as well as extra database related information read for each member table.

The member table configuration is instead maintained as `HTML` code (form hidden parameters) in the CGI (which replaces the `HTML` form invoking the servlet). For each possible member table (unused ones are automatically ignored) there is a sequence of parameters (indexed by a sequential number) which give member table name, region shape, database column names for `RA` and `Dec`, up to three size parameters used to control the size of the region, and an extra parameter which is an annotation appearing in the region table area of the tool. Also size parameters and the extra parameter are generated from the database, so they are `mysql` expressions used in the query.

**Table 6.** Sample HTML configuration for a member table

```

<input name="tab40" type=hidden value="CFHTLS_W1">
<input name="sha40" type=hidden value="circle">
<input name="ra40" type=hidden value="CFHTLS_W1_ra">
<input name="dec40" type=hidden value="CFHTLS_W1_dec">
<input name="p1_40" type=hidden value="greatest((26-greatest(i_old_MAG_AUTO,i_new_MAG_AUTO))/2,0.1)">
<input name="p2_40" type=hidden value="0">
<input name="p3_40" type=hidden value="0">
<input name="xtr40" type=hidden value="concat('I=',greatest(i_old_MAG_AUTO,i_new_MAG_AUTO),space(1),
'id=',CFHTLS_W1_id)">

```

Size parameters P2 and P3 are currently not used (they were used for ellipse-shaped regions, with P1, P2 being major and minor axis, and P3 the position angle).

An example of configuration items is reported in Table 6. The `tab` item is the member table name (here SF's CFHTLS W1 table); the `sha` item is the shape (e.g. circle, point, square, ellipse); `ra` and `dec` are the coordinates of the centre of the region, and usually correspond to the `mysql` name of a database column (but might be a complex expression). P1, P2 and P3 are size parameters. For a circle P1 is the radius in arcsec. In the example shown it is scaled from a magnitude. Usually  $(26 - mag)/2$  provides a sensible size, larger for brighter magnitudes. However it shall be protected for weak sources, forcing a minimum size (here 0.1"). The particular expression shown allows to use either `i` or `y` magnitudes, exploiting the fact that only one is defined at a time, and undefined is equal to -99). The `xtr` extra parameter is a label string made from an arbitrary `mysql` expression, here it concatenates the value of the `i` or `y` magnitude and the catalogue `id`. Since imbedded blanks screw up the `awk` script used by the CGI, the `mysql` `space()` function is used to generate a blank.

The combined result of the current colour and shape configuration is shown in Table 5.

Acknowledgements. I acknowledge the supply of the copious data products by Sotiria Fotopoulou.

## References

- Chiappetti, L., 2008a, More elements towards the XMDS full catalogue, The 2008 status report, XMM-LSS Internal Report N. 4-Mi ( Report IV)
- Chiappetti, L., 2008b, The INTERIM XMM-LSS catalogue, A status report, XMM-LSS Internal Report N. 5-Mi ( Report V)
- Chiappetti, L., 2010, Extending the 2XLSS catalogue, A status report in advance of publication, XMM-LSS Internal Report N. 7-Mi ( Report VII)
- Chiappetti, L., 2011, The ultimate 2XLSS deep and 10ks catalogues, The final report in advance of publication, XMM-LSS Internal Report N. 9-Mi ( Report IX)
- Chiappetti, L., 2013, The XXLN and XXLS catalogues, Preliminary release for internal use, XMM-LSS Internal Report N. 12-Mi ( Report XII)
- Chiappetti, L., 2014, Steps towards identification of XXL X-ray sources aka the Geneva exercise 0.0, XMM-LSS Internal Report N. 14-Mi ( Report XIV)