PyMidas–A Python Interface to ESO-MIDAS

Richard N. Hook

European Southern Observatory

Sami Maisala, Tero Oittinen, Marko Ullgren

Observatory, University of Helsinki

Kari Vasko, Ville Savolainen, Johan Lindroos

CSC, Finland

Matti Anttila, Otto Solin

Space Systems Finland Ltd.

Palle Møller, Klaus Banse, Michele Peron

European Southern Observatory

Abstract. Finland joined the European Southern Observatory in 2004, providing a contribution in kind of software expertise as part of its joining fee. This significant resource, called the Sampo project, will be devoted to exploring the options for the future of data reduction and analysis in an ESO context, to understanding user requirements and to performing a series of major pilot projects to investigate different technologies, approaches and architectures. The Sampo project[1] will run for three years and aims to prepare the ESO community for the data analysis and reduction challenges of the next decades.

The first major Sampo project is PyMidas, an interface from Python to the ESO-MIDAS data analysis and reduction system. This paper describes the motivation for this project, how it has been implemented and gives some examples of PyMidas in action.

1. Background and Motivation

Over the last few years the Python language has become very popular within astronomy. It is a fully-functional object oriented scripting language that is equally suitable for tiny test scripts and major software projects. It is also freely available and very stable and robust. The virtues of the language itself do not guarantee that it is ideal for astronomical applications, but fortunately

[1] `http://www.eso.org/sampo`

other developments have greatly helped its adoption. Firstly, it appears to be a language that both working scientists and professional software developers are happy to learn and use. Secondly, work by many groups is providing the additional packages, such as numarray and matplotlib, that are needed for truly useful astronomical application. Finally, connections to other major legacy astronomical software systems are emerging, most notably PyRAF where the IRAF 'cl' has been replaced by Python to produce a system with access to the full power of both the large base of IRAF applications and Python scripting.

These considerations led us to wonder how this approach could be extended in a European astronomical context. Although it has not been extensively developed for some years, ESO-MIDAS remains a widely-used general-purpose astronomical data processing system within the ESO community. ESO-MIDAS, like IRAF, represents a huge body of stable and tried-and-tested software built in a software environment that was designed several decades ago. As an initial Sampo project that would introduce the Sampo team to many aspects of astronomical software and also lead to a useful pilot product, we therefore decided to develop PyMidas—a Python interface to ESO-MIDAS.

2. Architecture and Implementation

One of the primary requirements for PyMidas was that it would use a standard ESO-MIDAS installation without the need to rebuild the applications base. It was also important that an experienced user of ESO-MIDAS could easily adjust to PyMidas without a significant learning curve. The architecture that has been adopted uses a background ESO-MIDAS process to which commands are sent from Python, via a small compiled 'C-layer'. This design allows Python scripting but also retains the ability to run ESO-MIDAS procedures, which are executed entirely within the background ESO-MIDAS process.

The architecture of PyMidas is shown in detail in Figure 1. Text files are used for part of the communication but sockets are also used. This model makes it easy to redirect all the text output from an ESO-MIDAS command back into a Python list object. Several utility methods are available within PyMidas to provide convenient access to the ESO-MIDAS keyword database, image pixel values, header keywords and entries in ESO-MIDAS tables.

PyMidas allows ESO-MIDAS commands to be invoked using three simple syntaxes. Firstly, there is a standard syntax for interactive use from Python. This requires a small change from the standard ESO-MIDAS syntax, to avoid the use of the '/' separator. This mode allows full use of Python facilities. In addition there is a 'mimic mode' that allows the use of exactly the same syntax as traditional ESO-MIDAS, at the expense of Python features. Finally, PyMidas may be imported as a standard Python package for use alongside other Python packages and systems such as PyRAF.

PyMidas was developed on Linux Fedora Core 3, but has also been successfully installed on other systems. The main requirements are Python 2.3 or later, along with normal packages such as Distutils, gcc and ESO-MIDAS 05SEP or later. The most recent version of ESO-MIDAS is currently required as some small changes for PyMidas were needed at the system level, but all future releases of ESO-MIDAS will automatically support PyMidas. More details

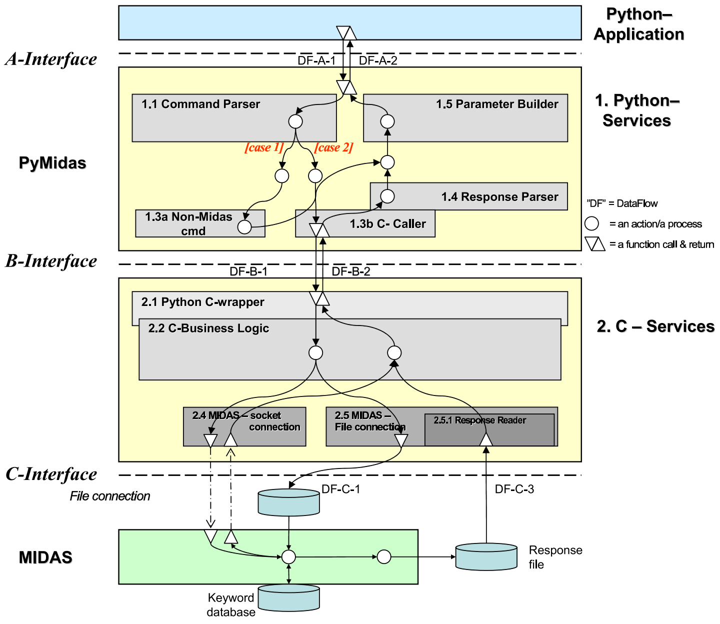


Figure 1. PyMidas Architecture

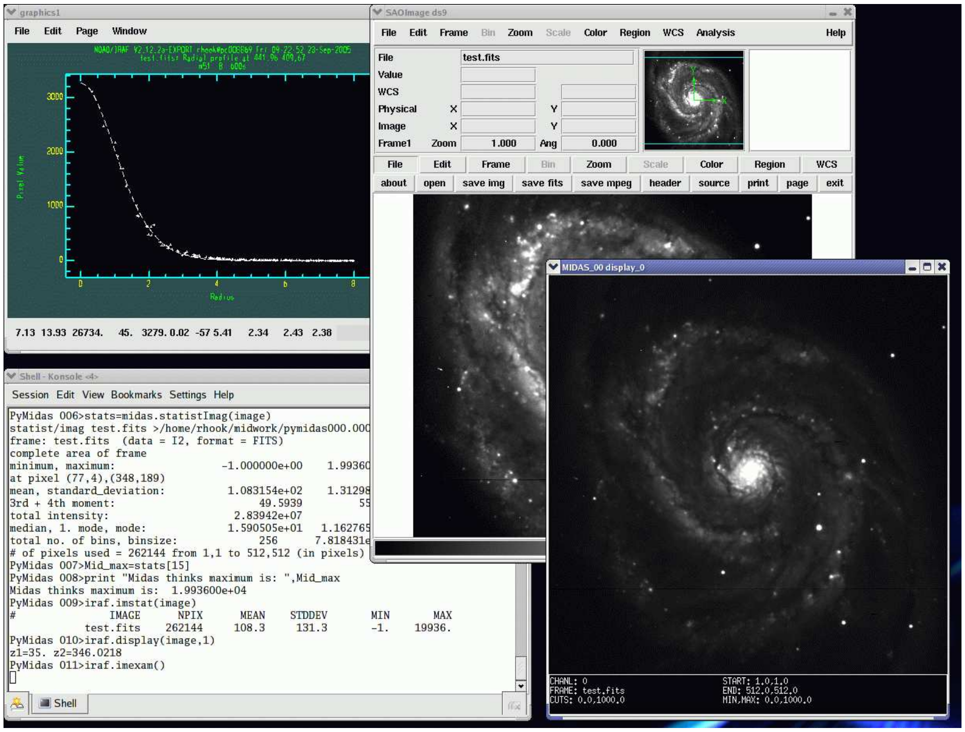
of requirements and installation instructions are available through the PyMidas web page[2].

3. Examples

Figure 2 shows a screenshot of an interactive session in which IRAF and ESO-MIDAS commands are mixed from PyMidas and an image displayed in both ds9 and with the ESO-MIDAS image display server. As a complementary scripting example the following short Python code snippet shows how PyMidas allows IRAF and ESO-MIDAS commands to be conveniently mixed, and information exchanged between them and Python internal variables. Some lines have been omitted for clarity:

```
# Load both PyMidas and PyRAF Python packages
from pymidas import midas
from pyraf import iraf

def run(image):

    # Get image stats from both ESO-MIDAS and IRAF
    midas.do('stat/image '+image)
    iraf.imstat(image)

    # Loop over a range of angles
    for angle in range(36):
```

```
# Rotate the image using drizzle in IRAF/STSDAS
iraf.drizzle(image,outdr,outweig=outwt,scale=1.0+angle/10., \
                kernel="turbo",rot=angle*5.0,outnx=512,outny=512)

# Add an offset value using ESO-MIDAS
midas.computeImag(outdr+' = '+outdr+'+'+str(angle*20.))

# Display the result with ESO-MIDAS
midas.loadImag(outdr,'cuts=0,1000')
```


graphics1
File Edit Page Window
Pixel Value
Radius
3000
2000
1000
0
0
2
4
6
8
SAOImage ds9
File Edit Frame Bin Zoom Scale Color Region WCS Analysis Help
File Edit Frame Bin Zoom Scale Color Region WCS
about open save img save fits save mpeg header source print page exit
MIDAS_00 display_0
Shell - Konsole <4>
Session Edit View Bookmarks Settings Help
Shell

```
NOAO/IRAF V2.12.2a-EXPORT rhook@pc006869 Fri 09:22:52 23-Sep-2005
test.fits: Radial profile at 441.96 409.67
m51 B 600s
```

```
7.13 13.93 26734.   45.  3279. 0.02 -57 5.41   2.34   2.43  2.38
```

File test.fits
Value
WCS
Physical X Y
Image X Y
Frame1 Zoom 1.000 Ang 0.000

```
PyMidas 006>stats=midas.statistImag(image)
statist/imag test.fits >/home/rhook/midwork/pymidas000.000
frame: test.fits  (data = I2, format = FITS)
complete area of frame
minimum, maximum:                 -1.000000e+00    1.99360
at pixel (77,4),(348,189)
mean, standard_deviation:          1.083154e+02    1.31298
3rd + 4th moment:                       49.5939         55
total intensity:                    2.83942e+07
median, 1. mode, mode:             1.590505e+01   1.162765
total no. of bins, binsize:              256      7.818431e
# of pixels used = 262144 from 1,1 to 512,512 (in pixels)
PyMidas 007>Mid_max=stats[15]
PyMidas 008>print "Midas thinks maximum is: ",Mid_max
Midas thinks maximum is:  1.993600e+04
PyMidas 009>iraf.imstat(image)
#              IMAGE      NPIX      MEAN    STDDEV        MIN        MAX
           test.fits    262144     108.3     131.3        -1.     19936.
PyMidas 010>iraf.display(image,1)
z1=35. z2=346.0218
PyMidas 011>iraf.imexam()
```

```
CHANL: 0                         START: 1.0,1.0
FRAME: test.fits                 END: 512.0,512.0
CUTS: 0.0,1000.0                 MIN,MAX: 0.0,1000.0
```

Figure 2. Screen-shot of an example session mixing ESO-MIDAS and IRAF commands from PyMidas

4. Distribution and Support

PyMidas has been developed as an experiment within the context of the Sampo project. As such it is not a formally supported piece of ESO software. The current release is available from the PyMidas web pages[2]. We plan to distribute it in future with ESO-MIDAS distributions, starting in autumn 2005, and as part of the ESO Scisoft software collection[3].

[2] http://www.eso.org/sampo/pymidas

[3] http://www.eso.org/scisoft