

Transcript of Calling ACR from Python

```
[noel@localhost ~/Desktop] $ python demo.py
>>> # parse the acr endpoint
.
>>> import sys
.
>>> import os
.
>>> prefix = file(os.path.expanduser("~/astrogrid-desktop")).next().rstrip()
.
>>> print prefix
http://localhost:8001/5742787e576d5bbe/
>>> #import the xmlrpc lib
.
>>> import xmlrpclib
.
>>> acr = xmlrpclib.Server(prefix + "xmlrpc")
.
>>> print acr
<ServerProxy for localhost:8001/5742787e576d5bbe/xmlrpc>
>>> # set up a pretty printer - for comfort of audience
.
>>> from pprint import pprint
.
>>> # see what the acr offers
.
>>> pprint(acr.system.listMethods())
.
['system.listMethods',
 'system.methodSignature',
 'system.methodHelp',
 'builtin.shutdown.halt',
 'system.browser.openURL',
 'system.browser.openRelative',
 'system.configuration.list',
 'system.configuration.listKeys',
 'system.configuration.setKey',
 'system.configuration.getKey',
 'system.ui.show',
 'system.ui.hide',
 'system.webserver.getUrlRoot',
 'astrogrid.community.login',
 'astrogrid.community.guiLogin',
 'astrogrid.community.logout',
 'astrogrid.community.isLoggedIn',
 'astrogrid.registry.resolveIdentifier',
 'astrogrid.registry.getRecord',
 'astrogrid.registry.getResourceData',
 'astrogrid.registry.search',
 'astrogrid.myspace.home',
 'astrogrid.myspace.node',
 'astrogrid.myspace.exists',
```

```
'astrogrid.myspace.getType',
'astrogrid.myspace.newFile',
'astrogrid.myspace.newFolder',
'astrogrid.myspace.readContent',
'astrogrid.myspace.copyContentToURL',
'astrogrid.myspace.copyURLToContent',
'astrogrid.myspace.getParent',
'astrogrid.myspace.getParentNode',
'astrogrid.myspace.getChildren',
'astrogrid.myspace.getChildrenNodes',
'astrogrid.myspace.refresh',
'astrogrid.myspace.delete',
'astrogrid.myspace.rename',
'astrogrid.myspace.move',
'astrogrid.myspace.changeStore',
'astrogrid.myspace.copy',
'astrogrid.applications.list',
'astrogrid.applications.fullList',
'astrogrid.applications.getInfo',
'astrogrid.applications.getToolTemplate',
'astrogrid.applications.validateTool',
'astrogrid.applications.validateToolFile',
'astrogrid.applications.listProvidersOf',
'astrogrid.applications.execute',
'astrogrid.applications.executeOn',
'astrogrid.applications.executeFile',
'astrogrid.applications.executeOnFile',
'astrogrid.applications.abort',
'astrogrid.applications.checkExecutionProgress',
'astrogrid.applications.getExecutionSummary',
'astrogrid.applications.getResults',
'astrogrid.jobs.list',
'astrogrid.jobs.getJob',
'astrogrid.jobs.getJobSummary',
'astrogrid.jobs.submitJob',
'astrogrid.jobs.submitJobFile',
'astrogrid.jobs.cancelJob',
'astrogrid.jobs.deleteJob',
'astrogrid.portal.openPage',
'userInterface.parameterizedWorkflows.run',
'userInterface.registryBrowser.show',
'userInterface.registryBrowser.hide',
'userInterface.myspaceBrowser.show',
'userInterface.myspaceBrowser.hide',
'userInterface.applicationLauncher.show',
'userInterface.applicationLauncher.hide',
'userInterface.jobMonitor.displayApplicationTab',
'userInterface.jobMonitor.displayJesTab',
'userInterface.jobMonitor.addApplication',
'userInterface.jobMonitor.show',
'userInterface.jobMonitor.hide',
'userInterface.jobMonitor.refresh',
'dialogs.resourceChooser.chooseResource',
'dialogs.toolEditor.editTool']
```

```
>>> #look at operations on CEA applications
```

```
.
```

```

>>> print acr.system.methodHelp('astrogrid.applications.list')
.
Module astrogrid
    Astrogrid Components
Component applications
    Query the application registry
Method list
    list names of registered applications
unknown : array
list of application ivorns
>>> pprint(acr.astrogrid.applications.list())
.
['ivo://org.astrogrid/GalaxeV',
 'ivo://org.astrogrid/Pegase',
 'ivo://org.astrogrid/SExtractor',
 'ivo://org.astrogrid/BPZ',
 'ivo://org.astrogrid/INTWideFieldSurveySIAP',
 'ivo://org.astrogrid/IoA/FIRST/object-catalogue/ceaApplication',
 'ivo://astrogrid.mssl/CallMakeMPEGFitsImages',
 'ivo://astrogrid.org/sec_dsa',
 'ivo://uk.ac.cam.ast/INT-WFS/merged-object-catalogue/ceaApplication',
 'ivo://uk.ac.cam.ast/INT-WFS/observation-catalogue/ceaApplication',
 'ivo://astrogrid.org/sec_dsa/ceaApplication',
 'ivo://astrogrid.mssl/msslxx-cea-application',
 'ivo://org.astrogrid/CrossMatcher',
 'ivo://org.astrogrid/CDS.XMatch',
 'ivo://astrogrid.mssl/solarimg_dsa',
 'ivo://org.astrogrid/HyperZ',
 'ivo://astrogrid.mssl/CallMakeMPEGFitsImage',
 'ivo://astrogrid.mssl/CallMakeMPEGFitsImg',
 'ivo://astrogrid.mssl/msslxx-cea-application/ceaApplication',
 'ivo://astrogrid.mssl/pal_esdo/ceaApplication',
 'ivo://astrogrid.mssl/pal_esdo/msslxx-cea-application',
 'ivo://astrogrid.mssl/pal_esdo/msslxx-cea-application/ceaApplication',
 'ivo://roe.ac.uk/DSA_SSA/ceaApplication',
 'ivo://mssl.ucl.ac.uk/SolarMovieMaker',
 'ivo://mssl.ucl.ac.uk/solarimg_dsa_app',
 'ivo://mssl.ucl.ac.uk/solar_events_dsa_app',
 'ivo://uk.ac.port.dsg/SExtractor',
 'ivo://uk.ac.port.dsg/HyperZ',
 'ivo://roe.ac.uk/DSA_6dF/ceaApplication',
 'ivo://mssl.ucl.ac.uk/solarimg_dsa/ceaApplication',
 'ivo://uk.ac.le.star/HelloWorldHttpApp',
 'ivo://mssl.ucl.ac.uk/solar_events_dsa/ceaApplication',
 'ivo://uk.ac.le.star/helloWorld',
 'ivo://mssl.ucl.ac.uk/SolarOverlay',
 'ivo://mssl.ucl.ac.uk/GenericMovieMaker',
 'ivo://mssl.ucl.ac.uk/solar_events_dsa_app/ceaApplication',
 'ivo://roe.ac.uk/DSA_2MASS/ceaApplication',
 'ivo://roe.ac.uk/DSA_USNOB/ceaApplication',
 'ivo://astrogrid.mssl/GrabSDOFiles']
>>> print acr.system.methodHelp('astrogrid.applications.getInfo')
.
Module astrogrid
    Astrogrid Components
Component applications

```

Query the application registry

Method getInfo

Return information about an application

applicationName : string

registry key of the application to query

for

unknown : string

Description of an application

```
>>> print acr.astrogrid.applications.getInfo('ivo://org.astrogrid/Pegase')
```

.
Application: org.astrogrid/Pegase

PEGASE is a code which computes the spectral evolution of galaxies. The evolution of the stars, gas and metals is computed according to user selected star formation laws and initial stellar mass function. The stellar evolutionary tracks extend from the main sequence to the white dwarf stage. The emission of the gas in HII regions is also taken into account. The effect of extinction by dust is also modelled using a radiative transfer code.

Parameter

Lower mass

Lower mass

name :LMASS

type :double

units :Solar Masses

accept encodings :standard

default value :0.1

Parameter

Stellar Winds

name :WINDS

type :boolean

accept encodings :standard

default value :Y

Parameter

Synthesized colours

name :COLOURS

type :binary

accept encodings :standard

Parameter

Global Extinction?

0: No extinction

1: Extinction for a spheroidal geometry

2: Extinction for a disk geometry: inclination-averaged

3: Extinction for a disk geometry: specific inclination

name :GLOBALEXTINCTION

type :integer

accept encodings :standard

default value :0

Parameter

Galactic winds

name :GALWIND

type :boolean

accept encodings :standard

default value :n

Parameter

SN Model (A/B/C)

model A, B or C of Woosley & Weaver

name :SNMODEL

type :text

accept encodings :standard

default value :B

Parameter

SubStellar fraction

Mass fraction of substellar objects formed (real in [0.,1.])

name :FRACSUB

type :double

accept encodings :standard

default value :0

Parameter

binary fraction

Fraction of close binary systems

name :BINFRAC

type :double

accept encodings :standard

default value :0.05

Parameter

Consistent Evolution

Consistent evolution of the stellar metallicity (y/n)?

name :CONEVOL

type :boolean

accept encodings :standard

default value :y

Parameter

Infall Metalicity

Metallicity of the infalling gas (mass fraction, real in [0.,1.])

name :INFALLMETAL

type :double

accept encodings :standard

default value :0

Parameter

Nebular Emission

name :NEBEMISS

type :boolean

accept encodings :standard

default value :n

Parameter

Infall Timescale

Infall timescale (Myr, real)

name :INFALLTIME

type :double

accept encodings :standard

default value :0.10000E+04

Parameter

Synthesized spectra

name :SPECTRA

type :binary

accept encodings :standard

Parameter

Initial Mass Function

1: IMF_Kennicutt.dat
2: IMF_Kroupa.dat
3: IMF_MillerScalo.dat
4: IMF_Salpeter.dat
5: IMF_Scalo86.dat
6: IMF_Scalo98.dat
7: lognormal IMF
8: Rana & Basu (1992)
9: Ferrini (1991)
 name :IMF
 type :integer
 accept encodings :standard
 default value :4

Parameter

Metallicity

Metallicity (mass fraction) of the ISM at t=0 (real in [0.,1.])
 name :METALICITY
 type :double
 accept encodings :standard
 default value :0.0

Parameter

Galactic wind age

Age of the galactic winds (Myr, real)
 name :GALWINDAGE
 type :double
 accept encodings :standard
 default value :0.20001E+05

Parameter

Star formation scenario

Type of star formation scenario?

- 2: file giving the SFR and the metallicity
- 1: file giving the SFR
- 0: instantaneous burst
- 1: SFR=p1 from t=0 to p2
- 2: SFR=p2*exp(-t/p1)/p1
- 3: SFR=(Mgas^p1)/p2

10 or more: code of the SFR law you have implemented in "spectra.f"

 name :SFSCENARIO
 type :integer
 accept encodings :standard
 default value :0

Parameter

Upper mass

Upper mass

 name :UMASS
 type :double
 units :Solar Masses
 accept encodings :standard
 default value :120.0

Parameter

Infall

 name :INFALL
 type :boolean
 accept encodings :standard

default value :y

Parameter

Stellar metallicity

(real in [0.,1.])

name :SMETAL

type :double

accept encodings :standard

default value :0.02

Interface simple

Inputs

LMASS max 1, min 1

WINDS max 1, min 1

GLOBALEXTINCTION max 1, min 1

GALWIND max 1, min 1

SNMODEL max 1, min 1

FRACSUB max 1, min 1

BINFRAC max 1, min 1

CONEVOL max 1, min 1

INFALLMETAL max 1, min 0

NEBEMISS max 1, min 1

INFALLTIME max 1, min 0

IMF max 1, min 1

METALICITY max 1, min 1

GALWINDAGE max 1, min 0

SFSCENARIO max 1, min 1

UMASS max 1, min 1

INFALL max 1, min 1

SMETAL max 1, min 0

Outputs

SPECTRAMax 1, min 1

COLOURSmax 1, min 1

>>> #hmm, lets look at something simpler.

.

>>> print

```
acr.astrogrid.applications.getInfo('ivo://org.astrogrid/INTWideFieldSurveySIAP')
)
```

.

Application: org.astrogrid/INTWideFieldSurveySIAP

This is part of the astrogrid CEA. It offers a CEA service to the INT WFS SIAPserver at Cambridge

Parameter

Position

Comma separated position of centre of field RA,DEC in decimal degrees

name :POS

type :text

subtype :none

units :String

accept encodings :standard

Parameter

Size

Comma separated size of field RA,DEC in decimal degrees

name :SIZE

type :text

```
    subtype :none
    units :String
    accept encodings :standard
```

Parameter

Format

e.g. image/fits - see SIAP definition

```
    name :FORMAT
    type :text
    subtype :none
    units :String
    accept encodings :standard
```

Parameter

ImagesTable

A VOTable of images

```
    name :IMAGES
    type :text
    subtype :none
    units :String
    accept encodings :standard
```

Interface Standard

Inputs

```
    POS max 1, min 1
    SIZE max 1, min 1
    FORMAT max 1, min 1
```

Outputs

```
    IMAGESmax 1, min 1
```

```
>>> #ok, lets call this app - ask the acr to generate a template call document
```

```
.
>>> print acr.system.methodHelp('astrogrid.applications.getToolTemplate')
```

```
.
Module astrogrid
```

```
    Astrogrid Components
```

```
Component applications
```

```
    Query the application registry
```

```
Method getToolTemplate
```

```
    Return a template tool document for an
        application
```

```
applicationName : string
```

```
name of the application to build a tool
```

```
for
```

```
applicationName : string
```

```
name of the interface to use - use 'default' for
```

```
default interface
```

```
unknown : string
```

```
Tool Document
```

```
>>> template =
```

```
acr.astrogrid.applications.getToolTemplate('ivo://org.astrogrid/INTWideFieldSurveySIAP','default')
```

```
.
>>> # lets see what we've got.
```

```
.
>>> from xml.dom.ext import PrettyPrint
```

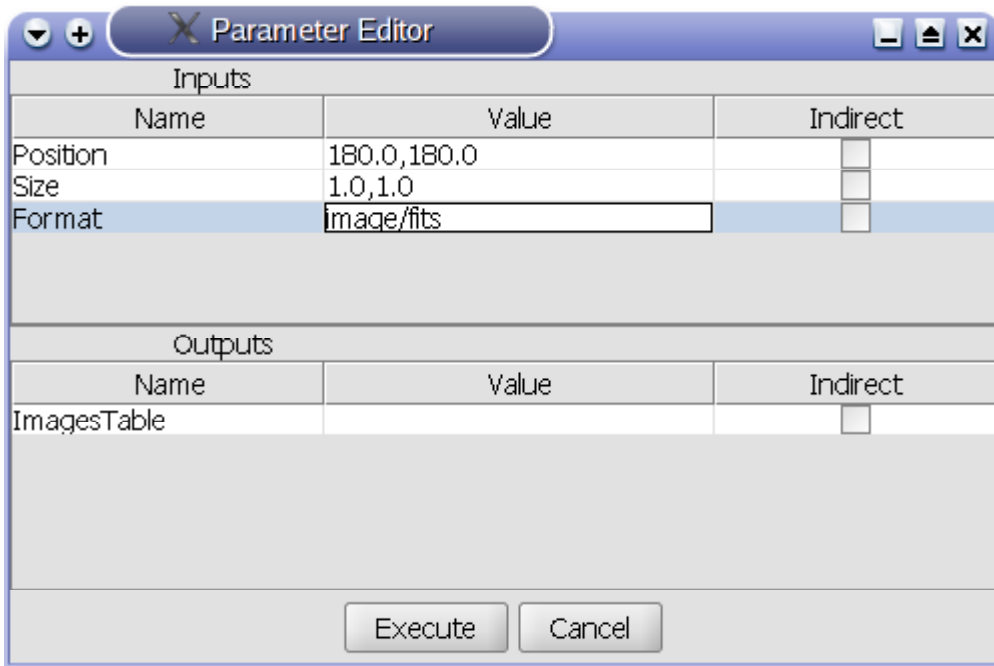
```
.
>>> from xml.dom.minidom import parseString
```



```

.
>>> PrettyPrint(parseString(template))
.
<?xml version='1.0' encoding='UTF-8'?>
<tool xmlns='http://www.astrogrid.org/schema/AGWorkflow/v1'
xmlns:ns1='http://www.astrogrid.org/schema/AGParameterDefinition/v1'
xmlns:ns2='http://www.astrogrid.org/schema/AGParameterDefinition/v1'
xmlns:ns3='http://www.astrogrid.org/schema/AGParameterDefinition/v1'
xmlns:ns4='http://www.astrogrid.org/schema/AGParameterDefinition/v1'
interface='Standard' name='org.astrogrid/INTWideFieldSurveySIAP'>
  <input>
    <parameter name='POS'>
      <ns1:value/>
    </parameter>
    <parameter name='SIZE'>
      <ns2:value/>
    </parameter>
    <parameter name='FORMAT'>
      <ns3:value/>
    </parameter>
  </input>
  <output>
    <parameter name='IMAGES'>
      <ns4:value/>
    </parameter>
  </output>
</tool>
>>> #don't really want to edit this by hand - use the tool editor dialogue
provided by ACR
.
>>> print acr.system.methodHelp('dialogs.toolEditor.editTool')
.
Module dialogs
  Reusable UI Dialogues
Component toolEditor
  Display a graphical editor for a CEA tool call document
Method editTool
  display editor
tool : string
Document to display in editor
unknown : string
An edited copy of this document
>>> tool = acr.dialogs.toolEditor.editTool(template)
.

```



```
>>> PrettyPrint(parseString(tool))
.
<?xml version='1.0' encoding='UTF-8'?>
<tool xmlns='http://www.astrogrid.org/schema/AGWorkflow/v1'
xmlns:ns1='http://www.astrogrid.org/schema/AGParameterDefinition/v1'
xmlns:ns2='http://www.astrogrid.org/schema/AGParameterDefinition/v1'
xmlns:ns3='http://www.astrogrid.org/schema/AGParameterDefinition/v1'
xmlns:ns4='http://www.astrogrid.org/schema/AGParameterDefinition/v1'
interface='Standard' name='org.astrogrid/INTWideFieldSurveySIAP'>
  <input>
    <parameter name='POS'>
      <ns1:value>180.0,0.0</ns1:value>
    </parameter>
    <parameter name='SIZE'>
      <ns2:value>0.01,0.01</ns2:value>
    </parameter>
    <parameter name='FORMAT'>
      <ns3:value>image/fits</ns3:value>
    </parameter>
  </input>
  <output>
    <parameter indirect='true' name='IMAGES'>
      <ns4:value>ivo://uk.ac.le.star/noelwinstanley#votable/demo.vot</ns4:value>
    </parameter>
  </output>
</tool>
>>> #run the app application
.
>>> print acr.system.methodHelp('astrogrid.applications.execute')
.
Module astrogrid
  Astrogrid Components
Component applications
```

Query the application registry

Method execute

Execute an application on a suitable server

tool : string

Tool document - describes the application and parameters to execute

unknown : string

A new execution identifier

```
>>> execId = acr.astrogrid.applications.execute(tool)
```

```
.  
>>> print execId
```

```
.  
ivo://org.astrogrid/RealHttpCEC#cea:zhumulangma.star.le.ac.uk/143.210.36.152/11  
16264446462:545046238
```

```
>>> # check progress of application
```

```
.  
>>> print acr.astrogrid.applications.checkExecutionProgress(execId)
```

```
.  
<?xml version="1.0" encoding="UTF-8"?>  
<execution-phase xmlns="http://www.astrogrid.org/schema/CEATypes/v1"/>  
>>>
```

```
.  
>>> # get a summary of the application
```

```
.  
>>>  
PrettyPrint(parseString(acr.astrogrid.applications.getExecutionSummary(execId))  
)
```

```
.  
<?xml version='1.0' encoding='UTF-8'?>  
<execution-summary-type xmlns='http://www.astrogrid.org/schema/CEATypes/v1'  
xmlns:ns1='http://www.astrogrid.org/schema/AGParameterDefinition/v1'  
xmlns:ns2='http://www.astrogrid.org/schema/AGParameterDefinition/v1'  
xmlns:ns3='http://www.astrogrid.org/schema/AGParameterDefinition/v1'  
xmlns:ns4='http://www.astrogrid.org/schema/AGParameterDefinition/v1'  
<application-name>org.astrogrid/INTWideFieldSurveySIAP</application-name>
```

```
<execution-id>cea:zhumulangma.star.le.ac.uk/143.210.36.152/1116264446462:545046  
238</execution-id>
```

```
<input-list>
```

```
<input indirect='false' name='POS' encoding=''>
```

```
<ns1:value>180.0,0.0</ns1:value>
```

```
</input>
```

```
<input indirect='false' name='SIZE' encoding=''>
```

```
<ns2:value>0.01,0.01</ns2:value>
```

```
</input>
```

```
<input indirect='false' name='FORMAT' encoding=''>
```

```
<ns3:value>image/fits</ns3:value>
```

```
</input>
```

```
</input-list>
```

```
<result-list>
```

```
<result indirect='true' name='IMAGES' encoding=''>
```

```
<ns4:value>ivo://uk.ac.le.star/noelwinstanley#votable/demo.vot</ns4:value>
```

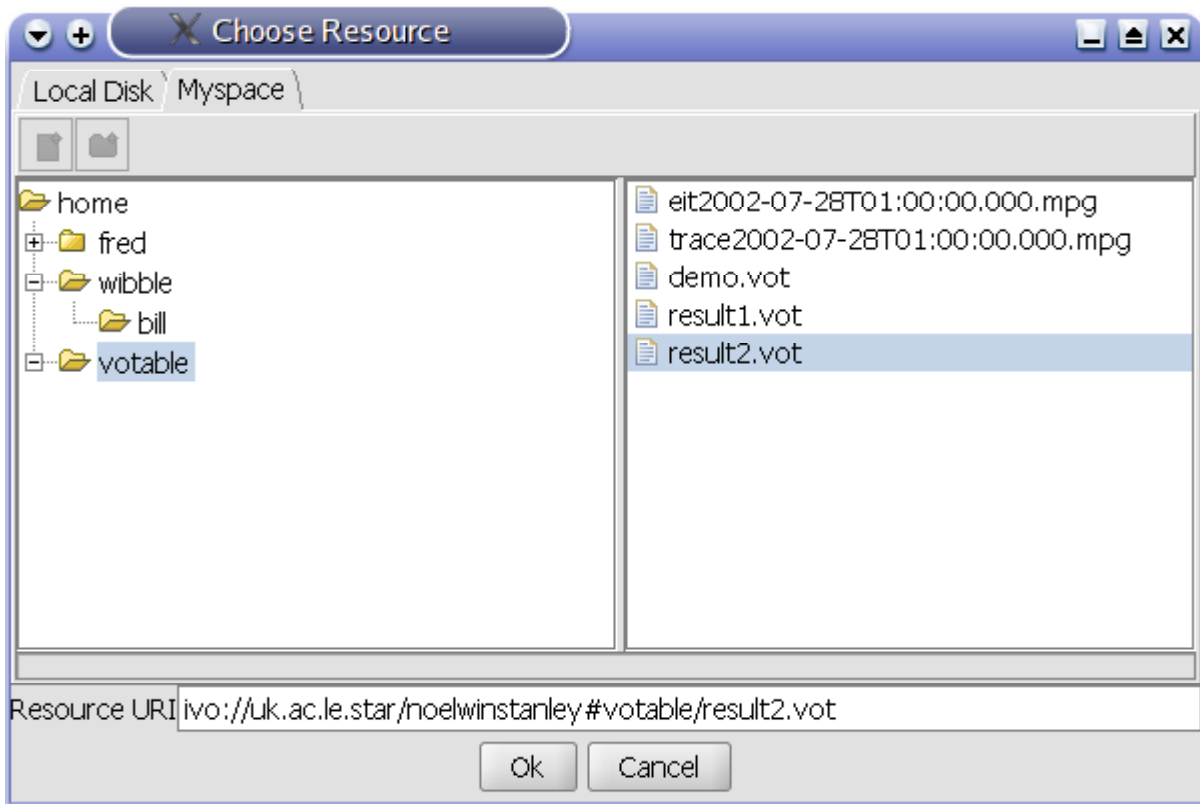
```
</result>
```

```
</result-list>
```

```
<status>COMPLETED</status>
```

```
</execution-summary-type>
```

```
>>>
.
>>> # check results in myspace - lets browse for the file
.
>>> ivorn = acr.dialogs.resourceChooser.chooseResource("", True)
.
```

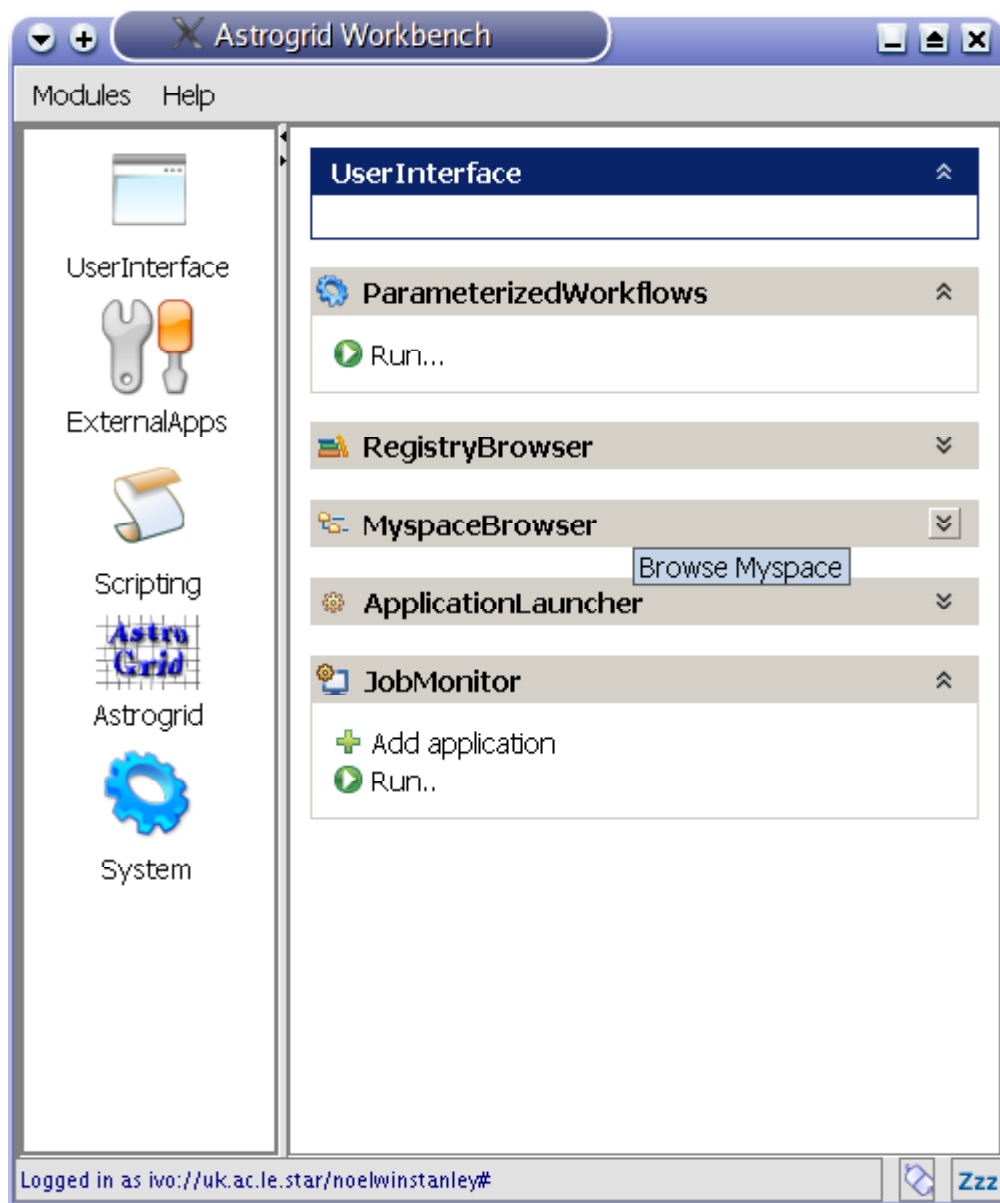


```
>>> print ivorn
.
ivo://uk.ac.le.star/noelwinstanley#votable/demo.vot
>>> dataURL = acr.astrogrid.myspace.readContent(ivorn)
.
>>> print dataURL
.
http://capc49.ast.cam.ac.uk/astrogrid-filestore/filestore/1818f80.103d11fefba.7
eb7
>>> # fetch this URL.
.
>>> from urllib2 import urlopen
.
>>> f = urlopen(dataURL)
.
>>> votable = f.read()
.
>>> PrettyPrint(parseString(votable))
.
<?xml version='1.0' encoding='UTF-8'?>
<VOTABLE xmlns='http://www.ivoa.net/xml/VOTable/v1.0'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://www.ivoa.net/xml/VOTable/v1.0
http://www.ivoa.net/xml/VOTable/v1.0'>
  <RESOURCE type='results'>
    <INFO name='QUERY_STATUS' value='OK' />
    <PARAM name='INPUT:POS' />
```

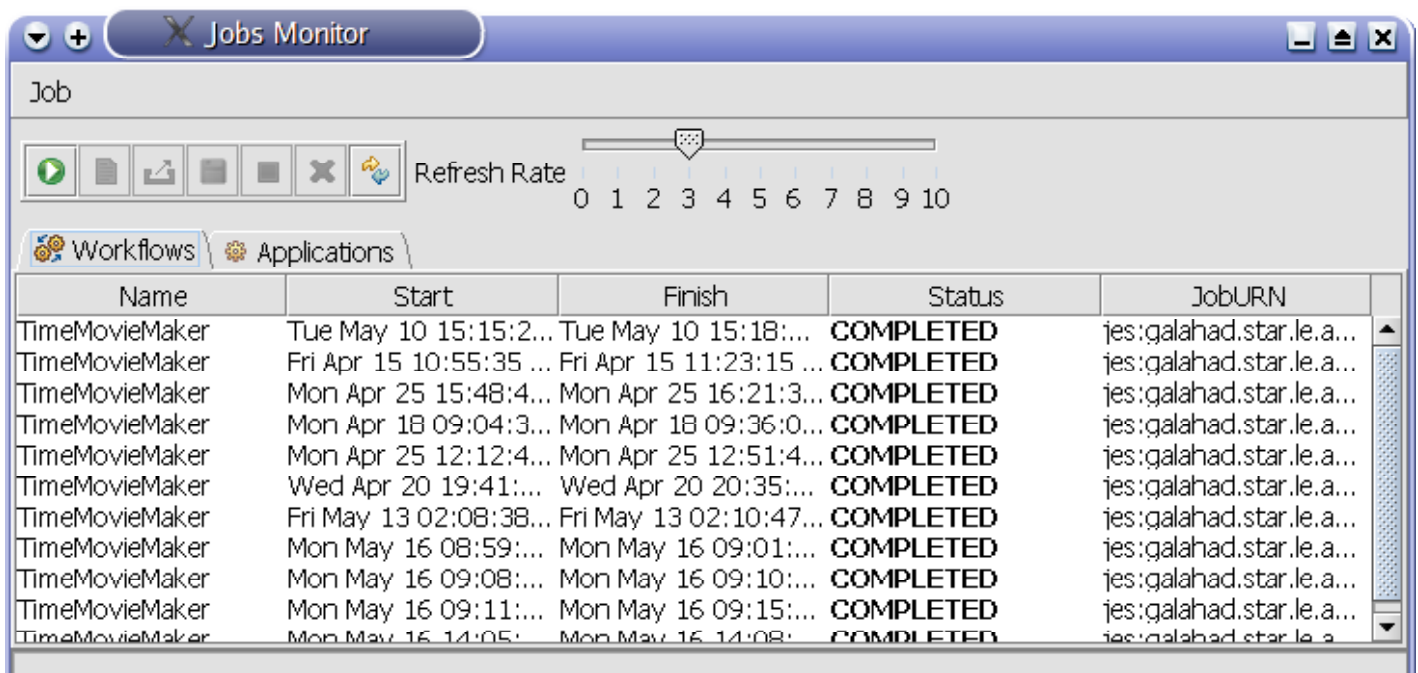
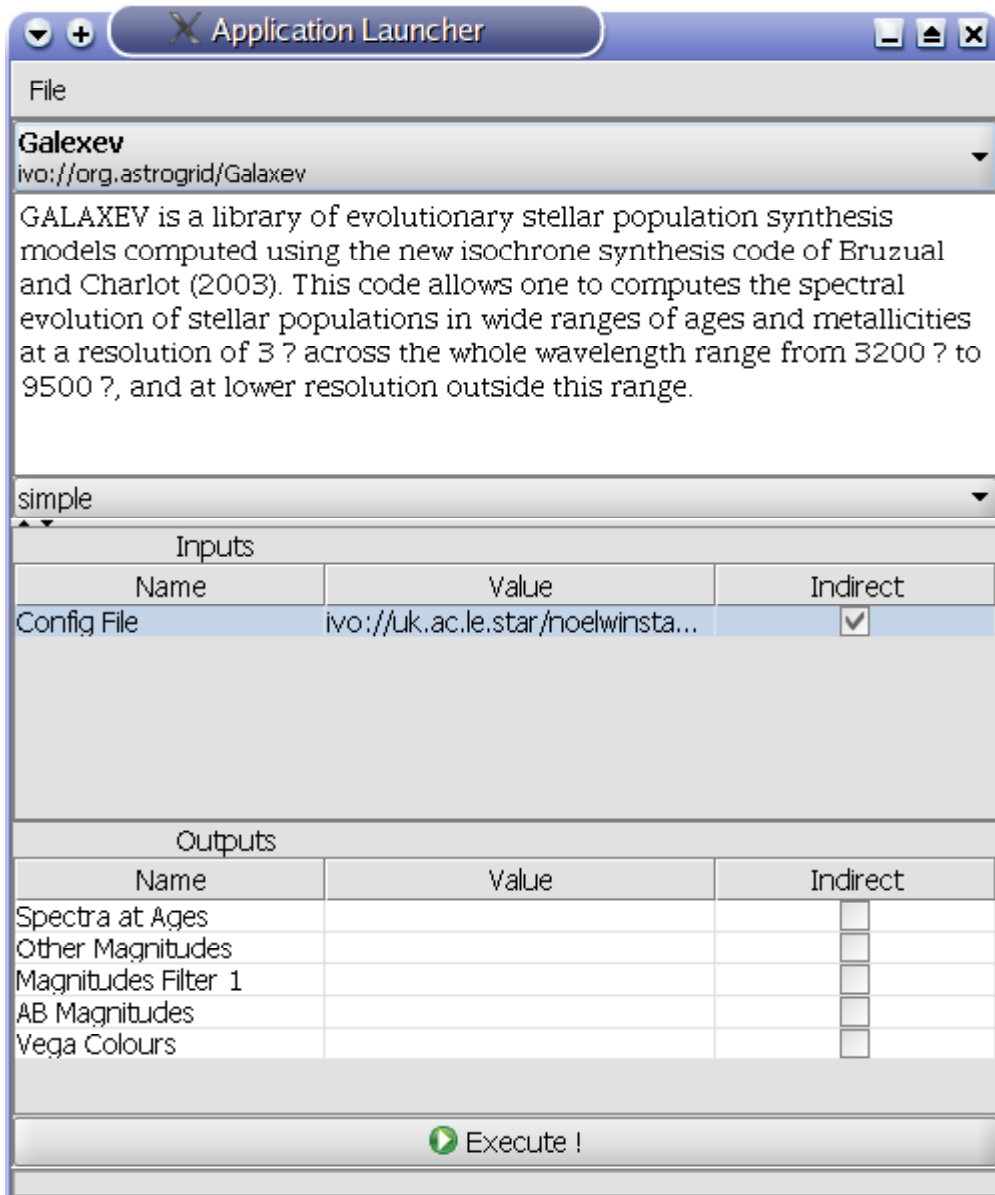
```

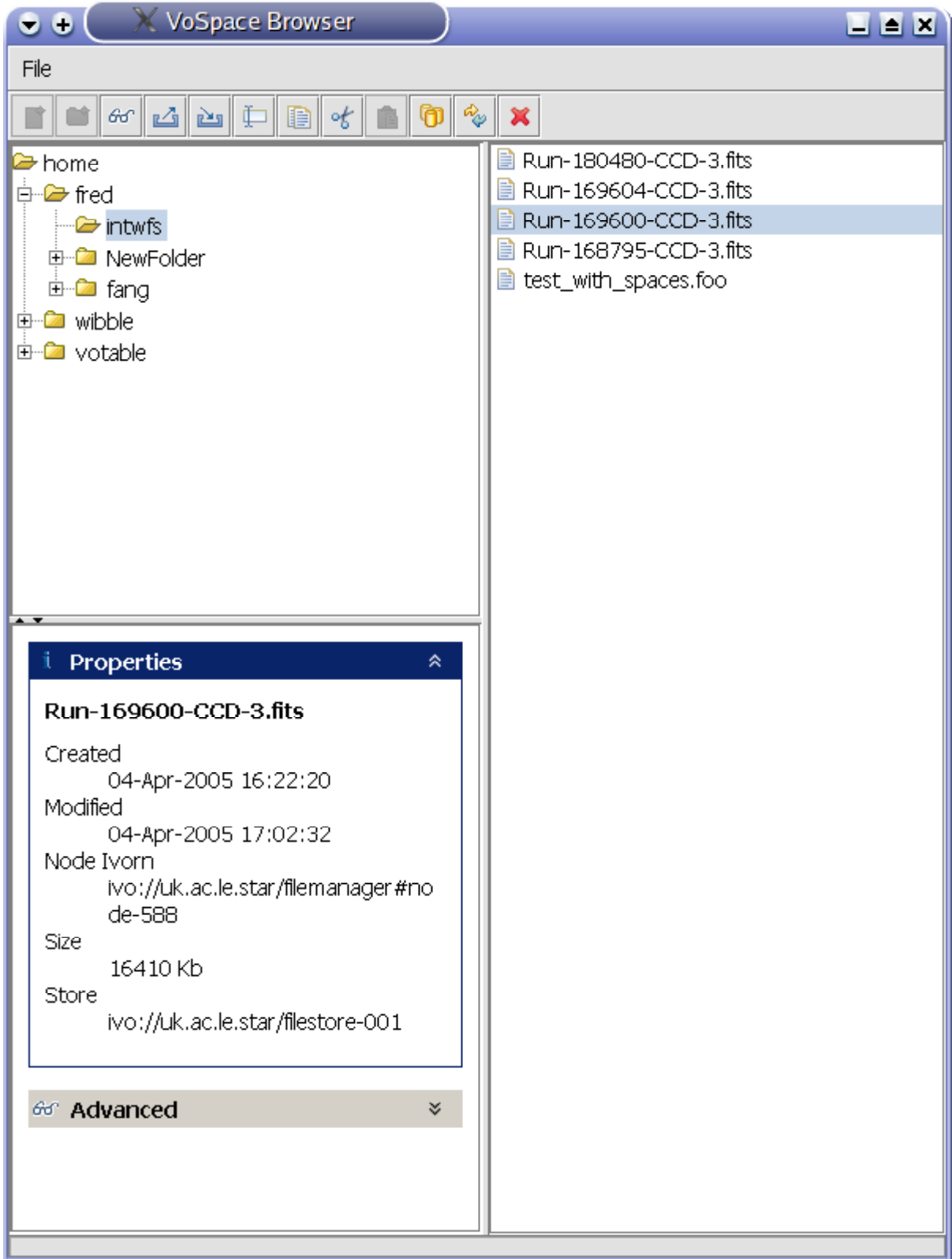
<PARAM name='INPUT:SIZE' value='0' />
<PARAM name='INPUT:FORMAT' value='ALL'>
  <VALUES>
    <OPTION value='image/fits' />
    <OPTION value='METADATA' />
    <OPTION value='ALL' />
  </VALUES>
</PARAM>
<TABLE>
  <FIELD datatype='char' ucd='VOX:Image_title' name='title'
arraysize='*' />
  <FIELD datatype='char' ucd='VOX:INST_ID' name='instrument'
arraysize='*' />
  <FIELD datatype='double' ucd='POS_EQ_RA_MAIN' name='ra' />
  <FIELD datatype='double' ucd='POS_EQ_DEC_MAIN' name='dec' />
  <FIELD datatype='int' ucd='VOX:Image_Naxes' name='naxes' />
  <FIELD datatype='int' ucd='VOX:Image_Naxis' name='naxis' arraysize='2' />
  <FIELD datatype='char' ucd='VOX:Image_Format' name='format'
arraysize='*' />
  <FIELD datatype='char' ucd='VOX:Image_AccessReference' name='accref'
arraysize='*' />
  <FIELD datatype='char' ucd='VOX:BandPass_ID' name='band-id'
arraysize='*' />
  <FIELD datatype='char' ucd='VOX:BandPass_Unit' name='band-unit'
arraysize='*' />
  <FIELD datatype='double' ucd='VOX:BandPass_HiLimit' name='band-upper'
unit='m' />
  <FIELD datatype='double' ucd='VOX:BandPass_RefValue' name='band-ref'
unit='m' />
  <FIELD datatype='double' ucd='VOX:BandPass_LoLimit' name='band-lower'
unit='m' />
  <FIELD datatype='char' ucd='VOX:Image_PixFlags' name='processing'
arraysize='*' />
  <DATA>
    <TABLEDATA>
    </TABLEDATA>
  </DATA>
</TABLE>
</RESOURCE>
</VOTABLE>
>>>
.
>>> # show the gui.
.
>>> acr.system.ui.show().

```



Workbench Screenshots





Appendices - Scripts

demo.py - run a script in 'demo' mode

```
#demo interpreter.
import code
import sys
script = file("/home/noel/Desktop/workbench-demo.py")
console = code.InteractiveConsole()
for l in script.xreadlines() :
    sys.stdout.write( ">>> " + l)
    raw_input(".")
    console.push(l)
```

workbench-demo.py - the demo script

```
# setting up - parse the acr endpoint
import sys
import os
prefix = file(os.path.expanduser("~/astrogrid-desktop")).next().rstrip()
print prefix
#import the xmlrpc lib
import xmlrpclib
acr = xmlrpclib.Server(prefix + "xmlrpc")
print acr
# set up a pretty printer - for comfort of audience
from pprint import pprint
# see what the acr offers
pprint(acr.system.listMethods())
#look at operations on CEA applications
print acr.system.methodHelp('astrogrid.applications.list')
pprint(acr.astrogrid.applications.list())
print acr.system.methodHelp('astrogrid.applications.getInfo')
print acr.astrogrid.applications.getInfo('ivo://org.astrogrid/Pegase')
#hmm, lets look at something simpler.
print acr.astrogrid.applications.getInfo('ivo://org.astrogrid/INTWideFieldSurveySIAP')
#ok, lets call this app - ask the acr to generate a template call document
print acr.system.methodHelp('astrogrid.applications.getToolTemplate')
template =
acr.astrogrid.applications.getToolTemplate('ivo://org.astrogrid/INTWideFieldSurveySIAP', '
default')
# lets see what we've got.
from xml.dom.ext import PrettyPrint
from xml.dom.minidom import parseString
PrettyPrint(parseString(template))
#don't want to edit this by hand - use the tool editor dialogue provided by ACR
print acr.system.methodHelp('dialogs.toolEditor.editTool')
tool = acr.dialogs.toolEditor.editTool(template)
PrettyPrint(parseString(tool))
#run the application
print acr.system.methodHelp('astrogrid.applications.execute')
execId = acr.astrogrid.applications.execute(tool)
print execId
# check progress of application
print acr.astrogrid.applications.checkExecutionProgress(execId)
# get a summary of the application
PrettyPrint(parseString(acr.astrogrid.applications.getExecutionSummary(execId)))
# check results in myspace - lets browse for the file
ivorn = acr.dialogs.resourceChooser.chooseResource("", True)
print ivorn
dataURL = acr.astrogrid.myspace.readContent(ivorn)
print dataURL
# fetch this URL.
from urllib2 import urlopen
f = urlopen(dataURL)
votable = f.read()
PrettyPrint(parseString(votable))
```

```
# audience.applause() - hopefully
```

```
# show the gui.  
acr.system.ui.show()
```