



# DiFX Python Interface

John Spitzak (USNO)



# Motivation

---

## USNO Effort to Automate Repetitive Goedesy Experiments

- Daily 1-hour “Intensives” are quite routine
- Python specification is clear
- Preempt external specification

## *guiServer* Has Many Capabilities

- Possibly well beyond command line
- Set up to handle remote connections
- Better half of the GUI client/server code

Something Different...



# What Is It?

---

## Series of Python Classes

- Some Complex
- Some Minor

## Example Python Programs

- Produced to test classes and demonstrate their use
- Possibly useful in their own right



# How To Run It

Server side: Requires *guiServer*

- Should be in DiFX bin directory
- Non-intrusive, (mostly) stable

```
difxmgr@node-001 DIFX-DEVEL ~-> guiServer&
[1] 27314
difxmgr@node-001 DIFX-DEVEL ~-> server at port 50200
```

Client side: Use One of the “Control” Classes

- Connect using *guiServer* host, port number

```
>>> import DiFXControl
>>>
>>> difx = DiFXControl.Client()
>>> difx.connect( "localhost", 50401 )
>>> difx.monitor()          # collect data from guiServer
>>>
>>> # do stuff
...
>>> difx.close()
□
```



# What Can It Do?

Simple Operations: *mv*, *rm*, *mkdir*, *rmdir*

```
>>> difx.mkdir( "/home/usno/users/difxmgr/foo" )
>>>
>>> difx.rmdir( "/home/usno/users/difxmgr/foo" )
>>>
>>> difx.rm( "/home/usno/users/difxmgr/foo", "-r" )
>>>
>>> difx.mv( "/home/usno/users/difxmgr/foo", "/home/usno/users/difxmgr/foo2" )
```

Transfer Data: *sendFile*, *getFile*

```
>>> junkData = "some stuff to send to a file\n"
>>> difx.sendFile( "/home/usno/users/difxmgr/foo", junkData )
29L
>>> retData = difx.getFile( "/home/usno/users/difxmgr/foo" )
>>> print retData
some stuff to send to a file
```



# What Can It Do?

List Directories (requires wait for response): `ls`

```
>>> dirList = difx.ls( "/home/usno/users/difxmgr/pp/*_0*.input", "-l" )
>>> for line in dirList:
...     print line
...
-rw-rw-r-- 1 difxmgr vlbi 12498 Nov  4 2014 /home/usno/users/difxmgr/pp/TestExperiment_01.input
-rw-rw-r-- 1 difxmgr vlbi 12498 Nov  4 2014 /home/usno/users/difxmgr/pp/TestExperiment_02.input
-rw-rw-r-- 1 difxmgr vlbi 12498 Nov  4 2014 /home/usno/users/difxmgr/pp/TestExperiment_03.input
-rw-rw-r-- 1 difxmgr vlbi 12498 Nov  4 2014 /home/usno/users/difxmgr/pp/TestExperiment_04.input
-rw-rw-r-- 1 difxmgr vlbi 12498 Nov  4 2014 /home/usno/users/difxmgr/pp/TestExperiment_05.input
-rw-rw-r-- 1 difxmgr vlbi 12498 Nov  4 2014 /home/usno/users/difxmgr/pp/TestExperiment_06.input
-rw-rw-r-- 1 difxmgr vlbi 12498 Nov  4 2014 /home/usno/users/difxmgr/pp/TestExperiment_07.input
-rw-rw-r-- 1 difxmgr vlbi 12498 Nov  4 2014 /home/usno/users/difxmgr/pp/TestExperiment_08.input
-rw-rw-r-- 1 difxmgr vlbi 12498 Nov  4 2014 /home/usno/users/difxmgr/pp/TestExperiment_09.input
```

Collect DiFX Message Traffic (requires a callback):

```
>>> def messageCallback( data ):
...     xmlDat = DiFXControl.parseXML( data )
...     print "got message type " + xmlDat.typeStr
...
>>> difx.addRelayCallback( messageCallback )
>>> difx.relayPackets()
>>> got message type DifxLoadMessage
got message type DifxLoadMessage
got message type DifxLoadMessage
got message type DifxLoadMessage
```



# What Can It Do?

## Obtain Last Run Status of a Job(s): *jobStatus*

```
>>> statusInfo = difx.jobStatus( "/home/usno/users/difxmgr/pp/TestExperiment_01.input", True )
>>> print statusInfo
('Thu Nov 12 14:55:11 2015', [( '/home/usno/users/difxmgr/pp/TestExperiment_01.input', ('Thu Nov 12 11:49:53 2015', 'MpiDone  ') ]])
```

## Manipulate Jobs: DiFXJobControl Class

```
>>> import DiFXJobControl
>>> difx = DiFXJobControl.Client()
>>> difx.connect()
>>> difx.monitor()
>>>
>>> difx.inputFile( "/home/usno/users/difxmgr/pp/TestExperiment_01.input" )
```

## Adjust .machines/.threads Files:

```
>>> difx.getMachines()
True
>>> difx.setHeadNode( "node-001" )
>>> difx.addDataSource( "node-002" )
>>> difx.addProcessor( "node-003", 10 )
>>> difx.defineMachines()
```



# What Can It Do?

## Start and Stop Jobs:

```
>>> difx.start()
>>> difx.stop()
>>>
```

## Collect Real-Time Monitor Data (bunch of code)

```
>>> def meanLagCB():
>>>     print "Delay:", difx.meanLagDelay, "SNR:", difx.meanLagSnr
>>>
>>> difx.startMonitor()
>>>
>>> monScans = difx.getMonitorProducts()
>>>
>>> # Pick a list of products...
>>> difx.requestProducts( productList )
>>> difx.runWithMonitor( True )
>>> difx.monitorDataCallbacks( None, None, None, None, None, meanLagCB )
>>>
>>> difx.start()
>>> difx.stopMonitor()
```





# What It Can't Do (Yet)

---

- Create New Jobs From .vex, .v2d Files
- Manipulate Mark5's
  - Send Mark5 Commands
  - View Mark5 Directories
  - Generate File Lists
  - Copy Mark5 Data



# Example Programs

## Simple Operations (Coding examples)

- *DiFXmkdir* - makes a new directory on server
- *DiFXls* - list directory contents on the server
- *DiFXgetFile* - get the contents of a file
- *DiFXsendFile* - create a file with specific content

## More Interesting

- *DiFXversion* - get DiFX version running on the server, view environment variables
- *DiFXMessages* - view DiFX UDP message traffic, filter by source, message type, and “identifier” (job, usually)
- *DiFXJobStatus* - parses .difxlog for a job, returns most recent status, can “monitor” in repeat mode



# Example Programs

---

## Kind of Nifty

- *DiFXBusy* - monitors what a correlator is working on, shows jobs running and % complete, nodes used
- *DiFXJobControl* - starts and stops jobs, can specify machines/threads, limited real-time monitoring capability



# Where To Get It

## Part of the DiFX Distribution

- `difx/applications/guiServer/trunk/python`
- Classes and example programs in:  
`.../python`
- Documentation:  
Generate with *doxygen* from:  
`.../python/docSrc`  
Top level in:  
`.../python/doc/index.html`