Robert C. Byrd Green Bank Telescope
NRAO Green Bank

Toney Minter and Jim Braatz

7th April 2003

GBT SOFTWARE PROJECT NOTE 26.1

GBT GO – IARDS Interface Specification

HTML version Available[1]

Contents

Abstract

This document describes the interface between the GBT Observer's Interface (GO) and the Interim Automatic Real-time Display software (IARDS).

[1] http://www.gb.nrao.edu/GBT/MC/doc/dataproc/gbtGOIARDSinterface/gbtGOIARDSinterface/gbtGOIARDSinterface.html

History

7th April 2003 Initial version of document.

1 glishd

The GO and IARDS packages are both written using glish scripts. It is a natural extension of these packages to use the glishd functionality of glish to allow GO and IARDS communicate. More information on glishd can be found in the "Glish Users Manual". glishd basically makes it possible to send a glish event from one glish session to another glish session which can even be on a different machine.

glishd will run as root on the machine victor. A setuid program is in place which can be used to start and stop glishd. This invocations of this are as follows (as gbtops on victor):

```
sudo /sbin/run-glishd -stop

sudo /sbin/run-glishd -status

sudo /sbin/run-glishd -start

sudo /sbin/run-glishd -help
```

Whenever a glish session "connects to" glishd, the root copy of glishd will spawn a copy of itself which is setuid to the owner of the glish session "connected to" glishd. This means that there can be several versions of glishd running at one time. However, the important one is the version owned by root. If it is not there then the GO – IARDS interaction will fail.

2 Available Machines

Only certain machines will be allowed to access glishd so that the GO–IARDS interaction will occur. The machines for which this is allowed can be found via the following Linux prompt:

```
ls /home/aips++/stable/keys/hosts/
```

If a machine is present in this listing then it can support the GO–IARDS communication. The GO session and the IARDS session will both have to be running on machines in this list for the real-time display to work properly.

3 GO Events

GO will send a single glish event to IARDS. The command to issue the glish event in GO will have the form

```
GOproc(project_id, scan, type)
```

where project_id provides the information on where to find the data (*i.e.* /home/gbtdata/project_id), scan is the first scan number of the observation for IARDS to display and type is either "point", "focus" or"other".

For types "point" and "focus" GO will expect data to be returned by IARDS for updating the local pointing coefficients or the subreflector Y axis focus position.

The name of the glish event which will allow communication between IARDS and GO will be go_aips2. The actual glish event passed from GO to IARDS will be called GOproc and will contain a record of the parameters passed along to the GOproc function in GO. An example of the issue of this glish event is as follows:

```
go_aips2->GOproc([project_id='/home/gbtdata/test',scan=1,type='other'])
```

4 IARDS Events

IARDS will send two events back to GO. These two events will contain solutions from IARDS on determining the local pointing coefficients and the subreflector Y direction focus position. Examples of these two events are as follows:

```
go_aips2->IARDSpoint([d_az=0.1, d_el=0.1, status=T])
```

and

```
go_aips2->IARDSfocus([y=0.1, status=T])
```

where d_az and d_el are the changes in the pointing coefficients in arcminutes, y is the subreflector focus position in millimeters and status indicates that the solution is valid (T) or invalid (F).