



## Prerequisites

- Go through the [system-setup](#) and make sure the system is operational. **Setup should be done in PFB mode.**
- **Make sure the Field System is configured for DBBC2 in PFB mode**
- Assumption: Fibre cable is connecting the FiLA10G to eth4 recorder1
- A single module is loaded and connected into slot 1 of recorder1
- **Make sure you have loaded the correct module intended for recording the GMVA session**

## FiLA10G Setup

- On the DBBC2 desktop double-click the icon labeled "**fila10g\_GMVAMode\_Mark6.bat**". This will setup the FiLA10G for recording on the Mark6 and will do the time-synchronisation. (The icon is a shortcut to the following file: C:\DBBC\bin\fila10g\_GMVAMode\_Mark6.bat).

DBBC2\_gmvamodemark6.png

- Confirm settings. In the terminal window that has opened the last page lists the output of the `sysstat` command. Inspect the output; in particular check that VDIF timesync is showing "yes" and "Output" is showing "started" (see image below).

```

System status:
Selected input      : vsi1
Input sample rate   : 64000000 Hz
USI input swapped  : no
USI input bitmask   : 0xFFFFFFFF
USI input width     : 32 bit
FPS count           : 11

TUG mode           : vsi-h
VDIF timesync      : yes
GPS receiver       : installed
Output             : started
Output 0 format    : vdif
Output 0 dest      : 172.16.2.1:46227
Output 1 format    : vdif
Output 1 dest      : 172.16.2.1:46227
Ethernet ARPs     : off

Selected USI output : vsi1-1
FiLA10G %
Press any key to continue . . .
  
```

- Confirm the time synchronisation as described [here](#).

## Mark6 module initialization

- **Skip this section if there is already recorded data on the module! The following steps will delete all data on the module!**
- log-into recorder1 as user oper: `ssh recorder1`
- execute: `da-client`
- in `da-client`:

```

mod_init=1:8:{VSN}:sg:new
on the module label
group=new:1
  
```

**replace {VSN}** with VSN as indicated

## Mark6 setup

- log-into recorder1: `ssh recorder1 -l oper -X`
- execute: `m6service_restart`
- execute: `da-client`
- in `da-client`:

```
mstat?1
```

make sure the group is NOT in "open" state. If it is in "open" state you must first do:

```
group=close:1
```

Otherwise continue with:

```
input_stream=add:DBBC2:vdif:5032:50:42:eth4:::1
input_stream=commit
group=open:1
mstat?1
```

The last statement should report the group to be in open:ready state, e.g.:

```
!mstat?0:0:1:1:MPIH%016/48008/4/8:8:8:45160:48008:open:ready:sg;
```

## Loading/starting the schedule

- log-into recorder1 as user oper: `ssh recorder1 -l oper -X`
- Download/copy the vex file into the directory: `/home/oper/vex`
- Translate the vex-file

```
cd /home/oper/vex
vex2xml.py -f {vexfile} -s PV
```

- Check the contents of the produced xml-file. It should not be empty and contain several scan entries, similar to this:

```
<experiment name="a90xx" station="PV" start="2015090003000" end="2015090032200">
  <scan experiment="a90xx" source="3C279" station_code="PV" start_time="2015090003000" duration="420"
scan_name="090-0030"/>
  <scan experiment="a90xx" source="3C279" station_code="PV" start_time="2015090004500" duration="420"
  <scan experiment="a90xx" source="3C279" station_code="PV" start_time="2015090031500" duration="420"
scan_name="090-0315"/>
</experiment>
```

**NOTE:** If the file is empty check that the station code given in the `vex2xml.py` command matches the station code for Pico Veleta of the vex-file.

- Start the schedule
  - Method 1 (preferred):

```
start_gmva.py {xmlfile}
```

where *{xmlfile}* is the xml-file produced by *vex2xml.py* (see above).

The script will start the schedule and in the scan gaps will display graphical information about the 16 PFB channels (bandpasses, and 2-bit statistics). The output is also stored in text-format under `/home/oper/GMVA/{code}`.

- Method 2 (backup):

```
M6_CC -f {xmlfile}
```

where *{xmlfile}* is the xml-file produced by *vex2xml.py* (see above)

## Field system

- log-into mrt-vlbi as user oper
- Download/copy the vex file into the directory: /usr2/sched
- run the vex file through *drudg*
- if not already running start the field system (click on the "field system icon" on the desktop)
- in the oprin window issue the command:

```
schedule={vexname}pv      (where vexname denotes the basename of the
vex file: e.g. c171a)
```

## End of session / Module change

When the session has ended or when a new module needs to be inserted:

- in da-client:

```
group=close:1
group=unmount:1
```

turn the module key off and remove the module.