DSM Hints

This page is a cheat sheet for the DSMs deployed with ISS for Perdigao, but most of it is general information which applies to any DSM. Two of the sites have a DSM.

Site	Hostname	IP address	Sensors
SODAR	dsm-sodar	192.1681.206	GPS, WXT (RS232 on ttyUSB3), Lufft (RS485 on ttyUSB1)
Sounding			GPS, Lufft

Logging In

The user account on a DSM is 'daq', ask Gary for the password. So you can ssh to a DSM like so:

> *ssh daq@192.168.1.206*

The DSM at the sounding site will be on the Ubiquiti tower network, so it can be reached from any host on that network. The SODAR DSM is connected to the Internet over a cell modem, so it can only be reached remotely through the SSH tunnel. To log into it directly, you have to be at the site and either on the TP-Link wifi or plugged into the ethernet switch in the DSM. Add the excerpt below to your ssh config file to use 'ssh dsm-sodar' to connect to it:

ssh config excerpt

```
Host dsm-sodar
User daq
HostName eol-rt-data.fl-ext.ucar.edu
HostKeyAlias dsm-sodar
Port 15422
```

Checking DSM Data

There are a few ways to check on the WXT data, so I'll just give examples of all of them. For reference, the WXT is plugged into port 3 of the SODAR DSM, which is device /dev/ttyUSB3.

Monitor serial data directly with rserial

Use the 'rs' script to tap the serial stream directly through NIDAS. The rserial program will print the messages received from the WXT while nidas still keeps recording the data.

```
rs 3
```

Run data_stats on the real-time stream

The DSM has a shortcut for the data_stats program called *ds*. By default the data_stats program connects to the real-time sample stream provided by the main *dsm* process, so it can be a convenient way to see which ports are receiving data and at what rates. It just keeps collecting samples until it is interrupted with Control-C. So it is not so useful with low-rate data, because you must wait until at least one sample should have been received to know if that sensor is working. You don't know if a one-minute sensor is sending messages unless you wait a full minute. If there is no line for a port in the output, then no samples were received on that port. If there is, then the line includes the dsm and sample ID, and those can be used to inspect the processed measurements with the *data_dump* command.

```
daq@dsm-sodar:~ $ ds
2017-05-05,00:56:55|NOTICE|parsing: /home/daq//isfs/projects/Perdigao/ISFS/config/perdigao.xml
^Creceived signal Interrupt(2), si_signo=2, si_errno=0, si_code=128
sensor                   dsm sampid    nsamps |------- start -------|  |------ end -----|    rate minMaxDT(sec)
minMaxLen
dsm-sodar:/dev/gps_pty0    5     10       101 2017 05 05 00:56:54.869  05 05 00:57:44.853    2.00  0.142
0.918   69   80
dsm-sodar:/dev/ttyUSB3     5     30         1 2017 05 05 00:57:04.725  05 05 00:57:04.725     nan  0.000
0.000  143  143
dsm-sodar:/dev/ttyUSB1     5    130         8 2017 05 05 00:57:00.003  05 05 00:57:40.003    0.18  5.000
10.000   69   78
IOException: inet:127.0.0.1:34182: recv: Interrupted system call
```

Show sample stats for the latest raw data file

NIDAS records the sensor data to a raw data file on the USB flash disk. While NIDAS is running, the disk is mounted on /media/usbdisk, and the data files are written to /media/usbdisk/projects/Perdigao/raw_data. So the command below uses the data_stats program to print sample statistics on all the sensors recorded in that data file:

Running data_stats on most recent raw data file

```
daq@dsm-sodar:~ $ data_stats `ls /media/usbdisk/projects/Perdigao/raw_data/dsm-sodar_2016* | tail -1`
2016-12-16,22:33:03|NOTICE|parsing: /home/daq//isfs/projects/Perdigao/ISFS/config/perdigao.xml
Exception: EOFException: /media/usbdisk/projects/Perdigao/raw_data/dsm-sodar_20161216_120000.dat: open: EOF
sensor dsm sampid nsamps |------- start -------| |------ end -----| rate minMaxDT(sec) minMaxLen
dsm-sodar:/dev/gps_pty0 5 10 75966 2016 12 16 12:00:00.046 12 16 22:33:02.808 2.00 0.142 0.930 69 80
```

The above shows that only the GPS messages are being received, at a rate of 2 Hz, with the most recent at 22:33. data_stats also gives the DSM and raw sample IDs for each sensor, so for the GPS messages the DSM ID is 5 and the sample ID is 10.

Dump individual messages

The data_dump program dumps individual messages from each sensor. Knowing the IDs given in data_stats, this command dumps all the WXT messages (ID 30) from the latest data file. Probably this would be piped into a pager like 'less':

```
data_dump -i 5,30 `ls /media/usbdisk/projects/Perdigao/raw_data/dsm-sodar_2016* | tail -1`
```

data_dump can also be used to dump samples immediately, as they are received by NIDAS, by tapping into a default network sample stream provided by NIDAS. Since this is real-time, nothing will be shown until a sample is received, so you have to wait for a minute before knowing that a 1-minute sample stream is not working.

```
data_dump -i 5,30
```

If you want to see all the sensors from data_dump, use '-i -1,-1', or data_dump -i '*,*', just remember to quote the asterisks.

Check that data files are recording

The **lsu** script to report the space available on the flash disk and also the last 10 data files. Run that script and check that the most recent data file corresponds to the current time.

Checking raw data file archive with lsu

```
daq@dsm-sodar:~ $ lsu
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       7.3G  187M  6.7G   3% /media/usbdisk
-r--r--r-- 1 daq eol  313400 Dec 13 16:37 dsm-sodar_20161213_120000.dat
-r--r--r-- 1 daq eol 4869400 Dec 14 00:00 dsm-sodar_20161213_161715.dat
-r--r--r-- 1 daq eol 6994554 Dec 14 10:34 dsm-sodar_20161214_000000.dat
-r--r--r-- 1 daq eol  856393 Dec 14 12:00 dsm-sodar_20161214_101718.dat
-r--r--r-- 1 daq eol 7933076 Dec 15 00:00 dsm-sodar_20161214_120000.dat
-r--r--r-- 1 daq eol 7932680 Dec 15 12:00 dsm-sodar_20161215_000000.dat
-r--r--r-- 1 daq eol 7932833 Dec 16 00:00 dsm-sodar_20161215_120000.dat
-r--r--r-- 1 daq eol 7555441 Dec 16 11:25 dsm-sodar_20161216_000000.dat
-r--r--r-- 1 daq eol  357247 Dec 16 12:00 dsm-sodar_20161216_111716.dat
-r--r--r-- 1 daq eol 7447951 Dec 16 23:25 dsm-sodar_20161216_120000.dat
daq@dsm-sodar:~ $ date
Fri Dec 16 23:25:45 UTC 2016
```

Note this doesn't indicate whether all the sensors are being recorded, only that at least one sensor has been recorded to disk recently.

Check time synchronization

The DSM should get its time from the GPS messages, synchronized to the PPS signal. The chronyd program manages the system time and keeps it in sync with GPS. So there are a couple things to check: GPS messages are being recorded and GPS is locked, and that chrony is working and synchronized.

cs is an alias for 'chronyc sources'. Use 'cs -v' to dump an explanation of the fields along with the stats:

Checking chrony sources with -v

```
daq@dsm-sodar:~ $ cs -v
210 Number of sources = 2
  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current synced, '+' = combined , '-' = not combined,
| /   '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||                                                /   xxxx = adjusted offset,
||         Log2(Polling interval) -.             |    yyyy = measured offset,
||                                  \            |    zzzz = estimated error.
||                                   |           |
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
#* PPS0                           0   4   377     9  +1139ns[+1211ns] +/-  391ns
#? NMEA                           0   4   377     9    +74ms[  +74ms] +/- 6256us
```

The asterisk in the PPS0 line is the most important, followed by the measured offset between GPS and system time inside the brackets. Typically it will be on the order of microseconds, as above where the offset is 1.21 microseconds.

rs G uses the rserial program to tap the GPS messages being recorded by NIDAS.

Checking GPS serial messages

```
daq@dsm-sodar:~ $ rs G
connecting to inet:localhost:30002
connected to inet:localhost:30002
sent:"/dev/gps_pty0
"
line="OK"
parameters: 4800 none 8 1 "\n" 1 0 prompted=false
$GPRMC,233927.00,A,3941.77412,N,00743.14572,W,0.020,,161216,,,D*65\r\n
$GPGGA,233927.00,3941.77412,N,00743.14572,W,2,08,1.48,373.5,M,49.2,M,,0000*46\r\n
```

The important part is the 'A' in the GPRMC line, meaning the time in that sentence is 'valid', whereas V indicates invalid.

Checking the DSM Configuration

The DSM data acquisition is configured with a NIDAS XML file. The XML contains a node for the particular dsm, where the hostname of the dsm matches the name in the dsm element. For example, below is the SODAR dsm entry. It may not match exactly what's on the DSM, depending on when the DSM was last updated. Also, the XML configuration file on the DSM may only contain an excerpt of the full configuration file, whatever is necessary to run on just that DSM.

/home/daq/isfs/projects/Perdigao/ISFS/config/perdigal.xml

```
     <site name="sodar" class="isff.GroundStation" suffix=".sodar">
        <dsm name="dsm-sodar" id="5" rserialPort="30002">
            <!-- GPS input is tee'd to this pseudo-terminal -->
            <serialSensor IDREF="Garmin_GPS_LITE" devicename="/dev/gps_pty0"
              id="10" readonly="true">
            </serialSensor>
        <serialSensor IDREF="LUFFT"
              devicename="/dev/ttyUSB1"
              id="130">
        </serialSensor>
        <serialSensor IDREF="WXT510" class="WxtSensor"
              devicename="/dev/ttyUSB3"
              id="30" suffix=".wxt">
        </serialSensor>
            <output class="RawSampleOutputStream">
                <socket type="server" port="30000" maxIdle="60" block="false"></socket>
            </output>
            <!-- Local Data Storage on DATAMNT, most likely media/usbdisk per dsm_env.sh -->
            <output class="RawSampleOutputStream">
                <fileset dir="$DATAMNT/projects/${PROJECT}/raw_data" file="${DSM}_%Y%m%d_%H%M%S.dat" length="
43200">
                    <mount dir="$DATAMNT"></mount>
                </fileset>
            </output>
        </dsm>
    </site>
```

The outermost **<site>** element contains a single DSM, which will usually be the case for ISS deployments. The **<dsm>** element has name *dsm-sodar*, and that matches the hostname of the DSM assigned to the sodar site. Then there are three sensors specified to be recorded by this DSM: GPS, LUFFT, and WXT510. Each of those names is a reference to a more complete **<sensor>** definition elsewhere, either earlier in the XML file or in a separate XML file called *sensor_catalog.xml*. The elements above only specify the device to which the sensor is attached, and the sensor ID required by NIDAS to differentiate samples from different sensors. The device names /dev/ttyUSB1 and /dev/ttyUSB3 correspond to ports 1 and 3 on the Raspberry Pi DSMs. Search for the LUFFT sensor reference to find out the rest of the configuration, such as the expected baud rate and other settings for serial sensors.

ISFS Man Page

There is a man page, 'man isfs'. It lists these shortcuts and a few others. Usually it is possible to get more detailed usage information by running the command without any arguments or by passing the -h option.