CMG CTBTO

Authenticating Seismic Array

Operator's guide

Part MAN-BSP-0003

Designed and manufactured by
Güralp Systems Limited
3 Midas House, Calleva Park
Aldermaston RG7 8EA
England

Issue C 2006-04-04

Table of Contents

1 Introduction

The CTBTO seismic station design is a complete solution for the seismic monitoring requirements of CTBTO, combining state-of-the-art authenticated digital borehole instruments with versatile networking capabilities, full AutoDRM and IMS command support and remote administration.

1.1 Overview

The station design is based on a single array of three sensors.

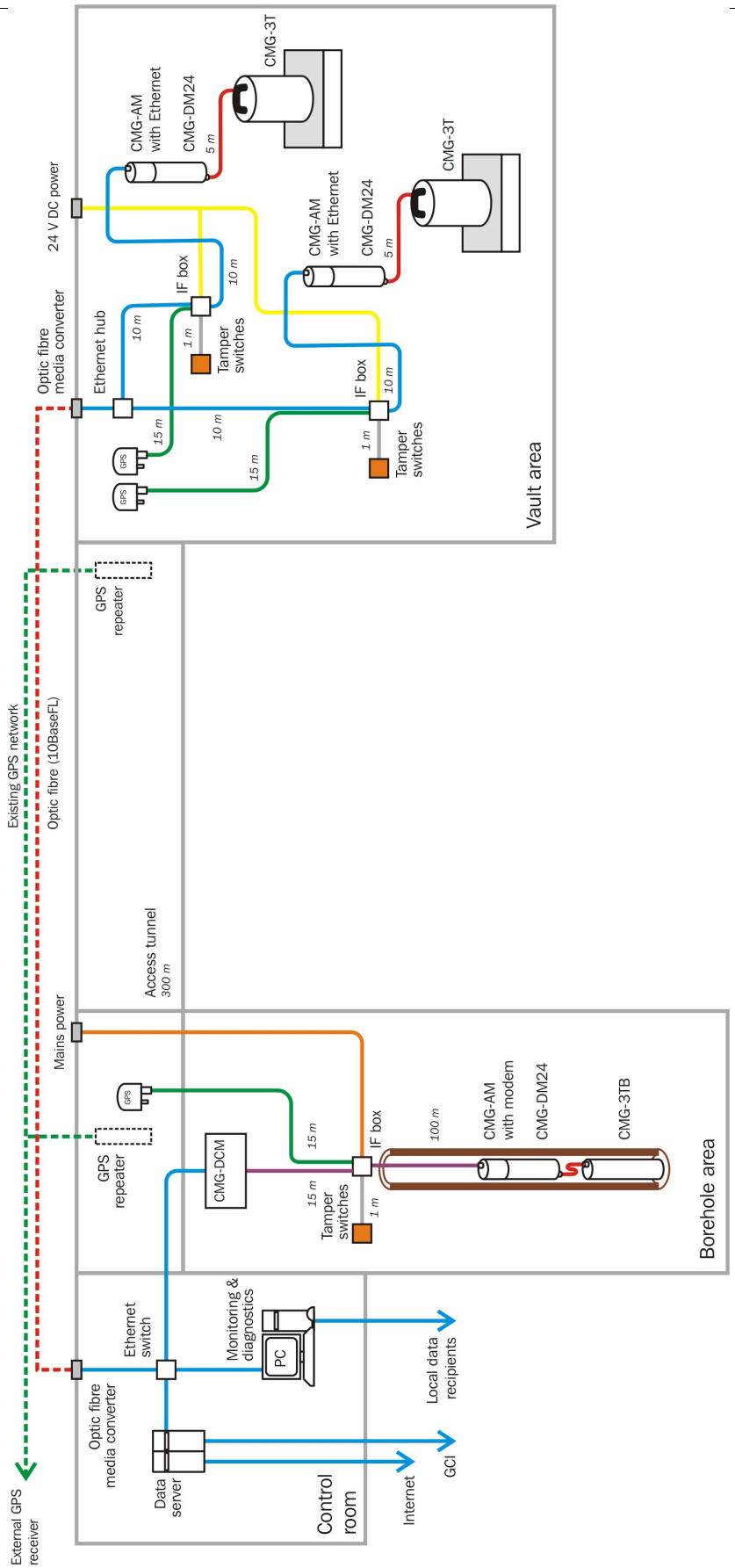
- A data centre is provided containing two Linux PCs, whose role is to collect and combine all incoming data and provide it to consumers and clients as configured. This data centre is connected to the Internet and GCI over separate network interfaces, as well as to the area network of the local institution. On the local side, a switch connects the two Linux PCs with the instrumentation.

- A 360 s CMG-3TB borehole seismometer, together with a combined downhole digitizer and Authentication Module (AM) unit, is connected to the switch at the data centre *via* Ethernet and modem links. This sensor receives power from the mains supply.

- Two 120 s CMG-3T weak motion vault seismometers, with identical digitizer-AM modules, are connected to the same switch *via* Ethernet and 10BaseFL optic fibre links. These sensors are supplied with 24 V DC power.

- GPS receivers are connected to all three instruments, sited close to existing GPS reradiating infrastructure within the building.

The physical layout of the system is depicted in the diagram below. Each solid line represents a separate cable, with cable lengths marked as appropriate. Dotted lines represent pre-existing infrastructure. “IF box” represents an interface or pit head box, supplied with each sensor, which contains power converters and network equipment appropriate for that sensor. For details of these, see chapter 6, page 35.


Vault area
CMG-3T
CMG-AM with Ethernet
CMG-DM24
5 m
24 V DC power
IF box
10 m
Optic fibre media converter
Ethernet hub
1 m
Tamper switches
15 m
GPS
GPS repeater
Existing GPS network
Optic fibre (10BaseFL)
Access tunnel
300 m
Mains power
CMG-DCM
100 m
CMG-3TB
Borehole area
Ethernet switch
Monitoring & diagnostics
PC
Local data recipients
Data server
Control room
Internet
GCI
External GPS receiver

1.2 Data flow arrangement

Signals from each instrument are digitized in the combined digitizer-AM module, authenticated at source with a Spyrus cryptographic token, and stored in Flash memory. Each AM has 1 Gb of memory, which corresponds to over 14 days of full-rate data. The AM formats the data into CD1.1 subframes and sends these to the data centre. Other data transfer methods may also be configured as required.

At the data centre, a dedicated server collates all the data from the instruments, forms full CD1.1 station frames, and transmits these to the NDC. A second Linux PC is provided for examining incoming data and monitoring the system, physically separated from the data server to ensure that critical services are maintained at all times. This PC is also used by the station operator to handle any AutoDRM requests which cannot be performed automatically.

The PCs at the data centre are part of the same local network as the borehole and vault installations, so from them you can connect to any of the instruments in the station to request data or update configuration settings. Depending on how the station is configured, administrators may also be able to log in securely over the Internet and change configuration settings. For more information on the PCs, see Chapter 7, page 38.

1.3 Authentication Modules

The power and flexibility of the station hinges on the Authentication Modules (AM) used within it. These units are extremely versatile networked devices which have built-in support for a wide range of commonly-used data formats and protocols. They run a specially-designed compact version of Linux, so you can log in over the network or a direct console connection to administer the system directly; they also provide a secure Web page interface, which administrators can use to configure the unit or any other instruments connected to it. Any of the AM units may be reconfigured at a later date to provide additional services as it becomes necessary.

If required, the AMs can be configured to serve data in Güralp Compressed Format (GCF), a highly compact and resource-efficient format suitable for storage and networking. With this feature enabled, any computer on the local network can communicate directly with the AMs, and retrieve and display data in real time using a suitable software application (such as Güralp Systems' own Scream! package.)

A second function of the AM units is to allow the operator to control

the pit installation remotely. The operator can

- cause the sensor to perform various tasks (such as locking, levelling and centring),
- configure the attached digitizer (*e.g.* to select appropriate stream rates and trigger criteria), and
- administer the AM itself (*e.g.* to run additional data services).

Any of these tasks can be performed on the AM's Linux console, or using an on-board Web server (see Section 2.4, page 13.) Much of the functionality can also be accessed through software such as Scream!.

1.4 The data server

This is a standard PC running CentOS Linux, whose purpose is to handle the data flow requirements of the array as a whole.

- It receives CD1.0 and CD1.1 subframes from the AMs and combine them into full station frames before sending them to the NDC and other data consumers as required.
- It receives e-mail for the station, including AutoDRM and IMS commands, and checks these for validity;
- It maintains a list of users and certificates, a set of Access Control Lists and a Certificate Revocation List, and checks the signatures on incoming e-mail against these to determine the rights of access.

Administrators can log in to this machine using *ssh* to perform system maintenance.

The data server is a key part of the station's infrastructure, so it is important that it remains available all times. Accordingly, it is not intended for day-to-day use by station operators. A second PC is provided for this purpose.

1.5 The monitor PC

This machine has identical hardware to the data server, and runs the same operating system. It runs all the system software not directly related to the station's function as a data producer:

- It maintains a Web page giving useful diagnostic information

about the station's status and performance, and allows operators to view historical status data.

- It runs Güralp Systems' Scream! software for additional diagnostic checks and to enable operators to view real-time data from the station; and

- It includes an e-mail client for the station operators, where messages are sent if they contain commands which must be performed manually, or if the station cannot parse them itself.

1.6 Using this manual

This manual describes how to install and set up each part of the array in a separate chapter. It is broken down as follows:

- The next chapter, *Using the system*, explains how to operate the array under normal conditions, including basic maintenance and diagnostics, once it is up and running.

- The following chapters focus in turn on the borehole and vault instrumentation, the digitizer-AM module, the interface unit and the data centre equipment. At the beginning of each chapter is an overview of the relevant component, including full installation instructions. Following this overview are sections providing technical details for the array component.

You should read this manual in conjunction with the operator's guides for each separate component of the system. The following additional documentation is supplied:

- CMG-3T weak motion seismometer;

- CMG-3TB borehole seismometer;

- CMG-DM24 broadband digitizer;

- CMG-AM authentication module;

- Scream! data visualisation software.

2 Using the system

2.1 Physical installation

Installation notes for each component of the array and data centre are provided in the relevant chapters of this manual. Please refer to those sections for details of how to install the hardware. In brief, you will need to

- install the different sensors to your satisfaction in as low noise environments as possible (see the *Installation* chapters in the relevant sensor manuals);
- find a location for the digitizer-AM unit and connect them to the sensors (see Chapter 5, page 26);
- connect each digitizer-AM unit to its interface box (see Chapter 6, page 35); and
- connect the interface box to its designed power source and network connection.

You will also need to refer to other sources to obtain information about the detailed arrangements for power distribution and networking at your site. This manual assumes that

- there exists a suitable source of mains power at the locations chosen for the interface boxes and the PCs at the data centre;
- this power source is suitably earthed, fitted with lightning and surge protectors, and backed up with UPS as required;
- there exists a local area network connecting the borehole and vault installations with the data centre;
- both computers at the data centre can route directly to the network and the NDC *without* using network address translation (NAT)—for example, because they have unique IP addresses on the Internet or GCI, or because they are on a LAN or WAN that includes the NDC;
- the AMs can make outgoing connections to the Internet (possibly using NAT) for software updates.

Once the pit installations and data centre are connected to the local network, the station is ready to operate.

2.2 Recording and storing data

Each AM in the system will automatically start recording data as soon as it is able to. Once a network connection is established to the installations, you can immediately use the full range of functionality of the instruments.

The data server is set up to fetch data from the AMs once it can find them on the network. Data from the whole array is combined and stored on its hard disk drives, from where it is forwarded to CD1.1 consumers (such as the NDC) as required. After a break in communications, any missing data is automatically backfilled either from the hard disk drive, or from the Flash memory of the AM units, depending on the location of the break. The AMs have sufficient Flash memory to store at least 2 weeks of full-rate data.

The data server as provided is already set up to format CD1.1 frames from the authenticated subframes that it receives from the pit installations. Should you need to alter the way this is done, please refer to the setup details given in Section 7.2, page 55.

2.3 Retrieving data from the array

There are three main ways you can get data from the array:

- as an NDC or other CD1.1 consumer, configured in the data server, which will automatically be sent CD1.1 station frames from the entire array;
- as an AutoDRM client, sending verifiable, S/MIME signed IMS messages to the data server, which will check their validity and integrity before sending them on to the AMs or station operator for action as necessary;
- as a Scream! client or other supported client type, receiving data direct from one or other of the AMs if it is configured to allow it.

Receiving data from the data server

Before the data server can send CD1.1 station frames to the NDC, it must be properly configured for its role in the array. Once it is configured, the data server will transmit CD1.1 station frames without any intervention from the station operator, including backfilling where necessary. If another client wishes to receive CD1.1 station frames

direct from the array, the data server will need to run a second concurrent CD1.1 transmitter process, with the appropriate configuration.

See Section 7.2, page 55, for details on how to set up the data server to send data to the NDC.

To form station frames, the data server also needs to be able to fetch correctly-formatted CD1.1 subframes from the AMs in the sensor installations. For this to work:

- the data server must be running a correctly configured `lxclient` process for each AM (see Section 7.2, page 54); and
- each AM must be configured to format the CD1.1 subframes (see !!!)

Requesting data from the AutoDRM

For security reasons, AutoDRM (Automatic Data Request Manager) messages are initially handled by the monitor PC, and not the data server. The messages should be sent to an address which will be delivered to the `autodrm` mailbox on the monitor PC.

The monitor PC will reject any AutoDRM messages which have not been cryptographically signed in S/MIME format with a recognized key. Depending on how it is configured, the monitor PC will either forward the verified AutoDRM messages for automatic processing (where possible) or place them in the operator's local mailbox for their attention.

Retrieving data directly from the pit installations

Each sensor installation has its own address on the local network. Requests sent to this address are handled by the AM in the digitizer package. Initially, the AM is configured to respond to CD1.1 lump requests from the data server, either directly (in the case of the vault installations) or through a managed modem link (in the case of the borehole.) Onward data flow, including backfilling, is managed by the data server according to the requirements of CD1.1 data transmission.

If it becomes desirable, the AMs may also be configured to act as servers for data in Güralp Compressed Format (GCF), a compact format suitable for transmitting over slow links. The option *datatransfer.screamserver* should be set to *on* to start the server. You may also want to set a maximum data rate for the GCF server in the option *datatransfer.screamserver.maxsamplerate*. See Section , page ,

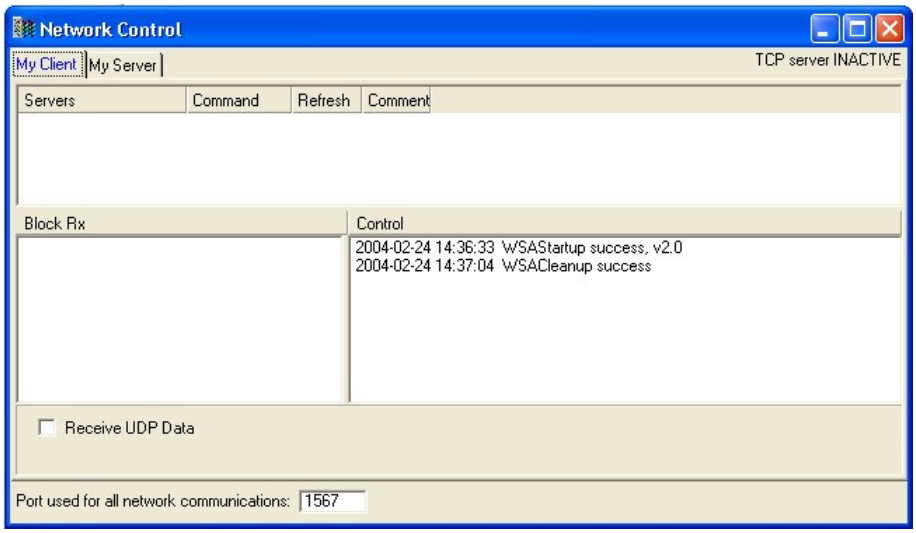
for full information.

Once an AM is running a Scream! server, any computer on the local network can connect to it using GCF-compatible software.

To connect to a sensor installation from a computer running Scream!, you should follow these steps:

1. Open Scream!'s main window, and right-click on the *Network* icon in the tree on the left.

2. Select *Configure...*. The *Network Control* window will open:



3. Click on the *My Client* tab at the top left of the window.

4. Ensure that the *Receive UDP Data* box is checked.

5. Right-click on the server list below it (showing *Servers, Command, Refresh,* and *Comment*).

6. Select *Add UDP Server...* and type the IP address and port number of the AM's GCF server into the box, separated with a colon. In their initial configuration, AMs serve GCF data on port 12345.

7. Click *OK* to add the server to the list. You should also see a message beginning *Added server...* appear in the *Control* pane.

8. Right-click on the name of the server in the list, and select *GCFPING*. A message beginning `CMD: GCFPING` will appear in the *Control* pane, signifying that a ping message was sent. The server should acknowledge the *GCFPING,* at which point a

second message will appear in the *Control* pane. If it does not, there is a problem with the UDP connection between your computer and the pit installation. See Section 2.5, page 15, for more suggestions.

9. Right-click on the name of the server in the list, and select *GCFSEND:B*. You should receive an acknowledgment message in reply, and after a while blocks of data should start appearing in the *Block Rx* pane.

10. Return to Scream!'s main window. The server should now be listed on the tree on the left under *Network*. Clicking on its name will list the streams that it is currently transmitting. Double-click on any of these streams to open a *WaveView* window on the stream.

For more information on how to use Scream! to display and manage data streams, please see Scream!'s own documentation or the extensive on-line help.

2.4 Configuring the instruments

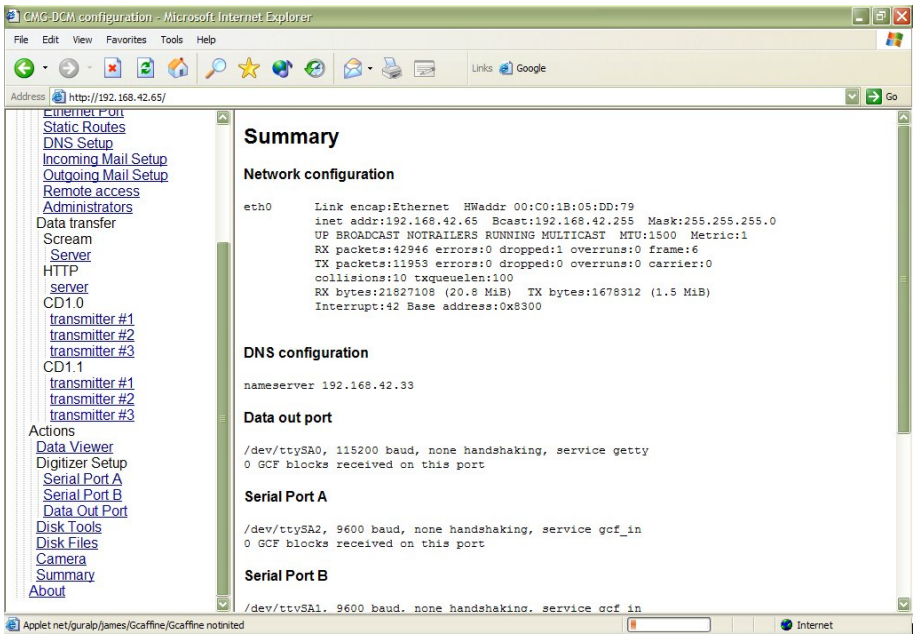
Each installation has three components which will need configuring: the AM, the digitizer, and the sensor. All three components can be set up through the AM's Web page interface. To access this from anywhere on the local network, open a web browser and navigate to the IP address of the AM. For example, if the AM has the address `10.82.0.1` you would enter

`http://10.82.0.1/` or `https://10.82.0.1/`

https is a secure variant of the *http* protocol, which ensures that network traffic cannot be read in transit. The AMs as supplied accept connections only over *https*, although this may be altered if required (see Section , page .)

Your browser will ask you for a username and password to access the Web configuration interface. As supplied, the AM units have a single user, `root`, with the password `rootme`. *This password is well known. You should change it as soon as possible.*

Once you have logged in, you will see a Web page similar to this:



The work area on the right displays a summary of the AM's current state, including a report on network traffic (produced by the Linux `ifconfig` command), details on the services provided by its internal serial ports, and current contents of the on-board Flash memory. See Section , page , for more information on this page.

The links in the menu on the left lead to various pages where you can change the installation's settings. All three components can be configured from here:

- The options under *Configuration* alter the behaviour of the AM itself, including communications, data protocols, and tamper evidencing.

- Under *Actions*, there will be a single link under *Digitizer Setup*. Clicking this link will cause the AM to retrieve the digitizer's configuration and display it as a Web form. You can change the configuration of the digitizer here, and upload it by clicking the *Upload* button.

- The same link provides buttons for controlling the sensor mass: *Unlock*, *Lock*, and *Centre*. You will need to *Unlock* each instrument before it will start measuring ground movements. See the chapter for the relevant sensor for more details. The sensor has no other configurable settings.

Many of the AM's own settings have been pre-set to suitable values, and you should not need to change them. In particular:

- The AM uses internal serial and Ethernet ports to communicate with the digitizer and the network. Altering the settings under *Serial port configuration* or *Network configuration* may make the AM unreachable over the network, or may prevent it from receiving data from the digitizer. You should review carefully any changes you make to these settings before committing them.

- You can disable the AM's *http* and *https* servers using the links under *Data transfer*. This is not recommended, since the data server uses *http* to request CD1.1 subframes from the AM. Disabling the *http* and *https* servers will thus prevent the subframes from being transmitted, as well as removing the Web configuration interface.

When you have changed any of the settings on a page, you must click the *Save settings* button at the bottom of that page. If you switch to another page without clicking *Save settings*, you will not have committed your changes, and may lose the values you have filled in.

The password for `root` can be changed on the *Administrators* page.

2.5 Diagnostics and troubleshooting

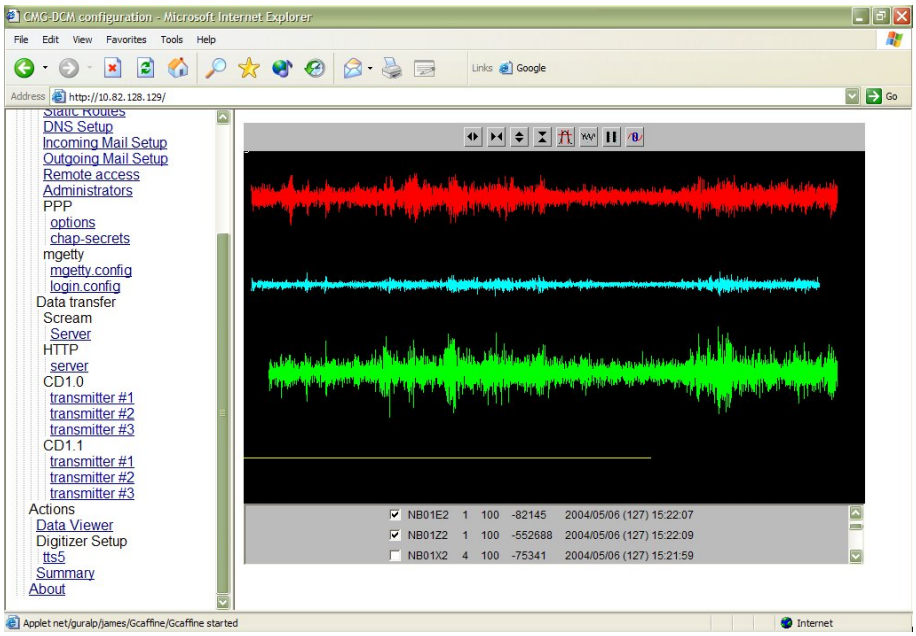
Troubleshooting the installations

The on-board Web server of each AM includes a Java applet which you can use to check that seismic data is being correctly measured. You will need a Java-compatible browser to use this feature.

To access the data viewer, log in to the Web interface of one of the AMs and select *Data viewer* beneath *Actions* in the left-hand pane. You may need to type in your username and password again to give the applet access permissions.

The data viewer is not intended to be a fully functional data visualisation tool. Scream! and similar software packages offer a full range of facilities for displaying and manipulating seismic data, either as it arrives or from stored data files. However, you can use the data viewer to verify that a pit installation is working correctly, either from a directly-connected laptop computer or from anywhere on the local network.

The main part of the display shows a selection of the data streams being recorded. The icons at the top allow you to zoom in and out, pause, or find the data's zero point, whilst the bottom section provides a list of checkboxes for you to select which streams to display.



If you cannot see any data in the Data Viewer, you should check first that

- some streams have been checked in the lower portion of the applet, and
- the data has been offset correctly. You can automatically zero the streams by clicking on the rightmost icon (depicting a “0” and a sine wave) in the icon bar. Otherwise, you can zoom out until the streams are visible.

If you have checked these and still see no data appearing, or if no streams appear in the lower portion of the applet, the AM is not receiving any data from the digitizer. Contact Güralp Systems for assistance.

If you see a constant, flat-line signal, and zooming into the Y axis does not show any change, it may be that the sensor mass is still locked. You can unlock the sensor mass from the *Digitizer Setup* page (accessible from a link near the bottom of the left-hand menu.) Before doing this, you should check carefully that the instrument is not producing data, since it cannot take measurements during the unlocking procedure.

Tamper switches

The AMs constantly monitor the state of the tamper switches in the borehole installations. This state can be retrieved by visiting the *Summary* section of the AM web page, which displays a table of

current tamper information. The information is also sent out as part of the CD1.1 transmission, according to the standard.

```
Input  State   Last Closed (Low)          Last Open (High)
    0   open    (never)                   Wed Jul  7 16:04:48 2004
    1   open    (never)                   Wed Jul  7 16:04:48 2004
    2   open    (never)                   Wed Jul  7 16:04:48 2004
    3   open    (never)                   Wed Jul  7 16:04:48 2004
    4   open    (never)                   Wed Jul  7 16:04:48 2004
    5 closed  Wed Jul  7 16:04:48 2004    (never)
    6   open    (never)                   Wed Jul  7 16:04:48 2004
    7 closed  Wed Jul  7 16:04:48 2004    (never)
    8 closed  Wed Jul  7 16:04:48 2004    (never)
    9 closed   (never)                    (never)
   10 closed   (never)                    (never)
   11 closed   (never)                    (never)
   12 closed   (never)                    (never)
   13 closed   (never)                    (never)
   14 closed   (never)                    (never)
   15 closed   (never)                    (never)
```

The columns in the table give, in turn, the current state of each switch and when it was last observed to be in each state. The above example shows a typical reading for a set of 9 tamper switches that have not triggered: the normally-closed switches show the current time under “Last Closed”, and the normally-open switches under “Last Open”. All other fields show `(never)`, indicating that no switches have been observed in the “wrong” state since the AM was last booted.

As shipped, tamper lines 7 and 8 are connected to a switch and opto-sensor located within the digitizer-AM module. They set the *DIGITIZING EQUIPMENT OPEN* and *EQUIPMENT HOUSING OPEN* flags in CD1.1. Tamper line 5 is connected to a switch inside the interface box, and sets the *EQUIPMENT HOUSING OPEN*flag.

In addition, the interface box provides a *TAMPER* connector which can be attached to up to 5 further switches (*Tamper 0 – 4*). Of these, one switch (*Tamper 0*) is programmed to set the *VAULT DOOR OPEN* flag in CD1.1.

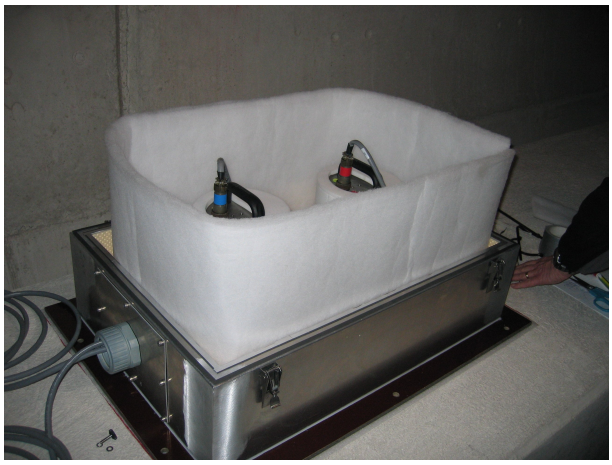
All tamper switches are set to trigger if open (*i.e.* active high.)

3 Vault installations

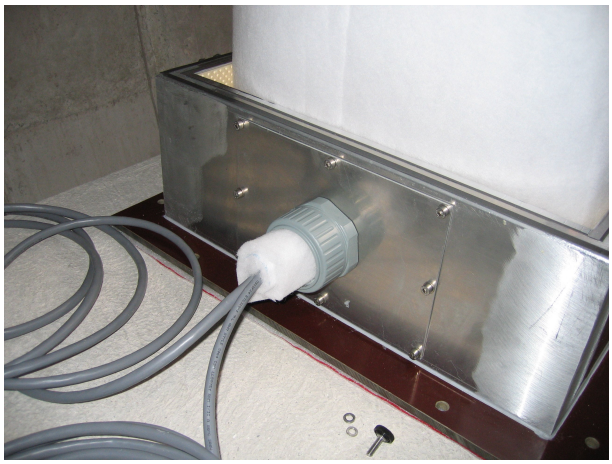
The two vault installations are co-located on a seismic pier at the end of the access tunnel. One CMG-3T seismometer for each installation is housed within an protective box. Cables from the seismometers exit the box through a single port.

Each instrument is additionally protected by a polystyrene sheath.

A further layer of polystyrene insulation within the box serves to exclude air currents. This is covered in the final installation.

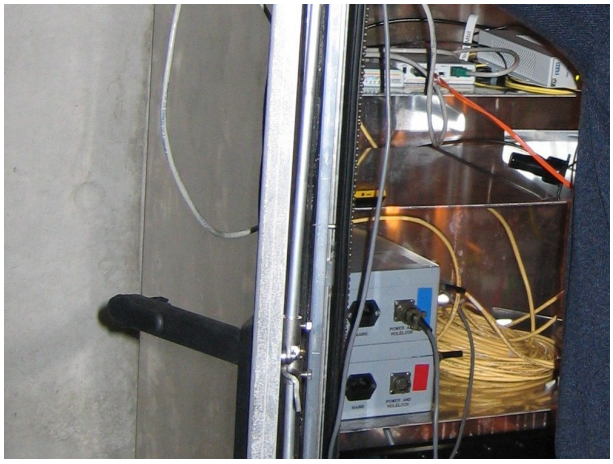


Finally, the empty space in the cable port is plugged with polystyrene insulation.



Nearby, a combined CMG-DM24 and AM unit for each installation takes analogue signals from the instrument, digitizes these, and authenticates the resulting data.

The output from each DM/AM unit is connected to an interface box, which exposes the power and console ports and sends data on to the data centre.



For general sensor installation instructions, operational notes and technical information, please see the CMG-3T manual, supplied separately.

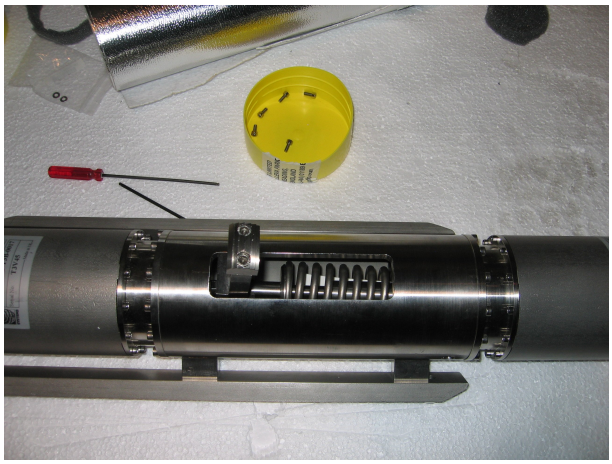
3.1 Testing the vault installations

You can test the DM/AM module on-site using a laptop computer, by connecting a Güralp serial console cable to the *DM24 CONSOLE* or *AUTH CONSOLE* connector on the interface box. The Linux program `minicom`, or `hypertrm` on Microsoft Windows, can be used to communicate over a direct serial link

4 Borehole installation

The borehole installation is based on a CMG-3TB borehole seismometer with single-jaw hole lock.

The hole lock unit is supplied with two metal skids, which assist the hole lock arm in gripping the inner wall of the borehole.

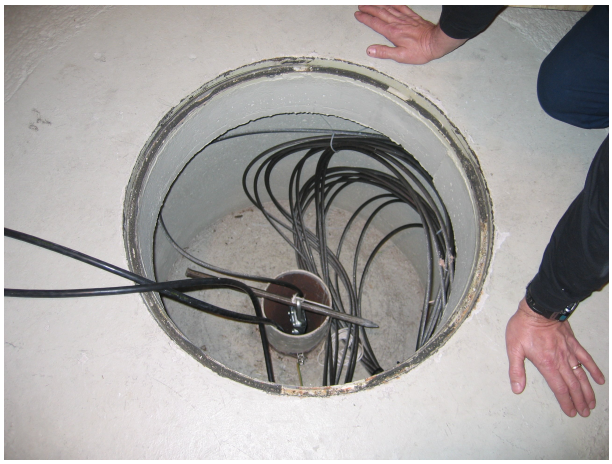


Suspended above the instrument on the same lifting cable is a DM/AM module of similar design to those in the vault (see below.) A strain relief unit below this module prevents the data cable from being placed under tension.

Before installation in the borehole, the sonde sections of the instrument were sheathed with polystyrene insulation.

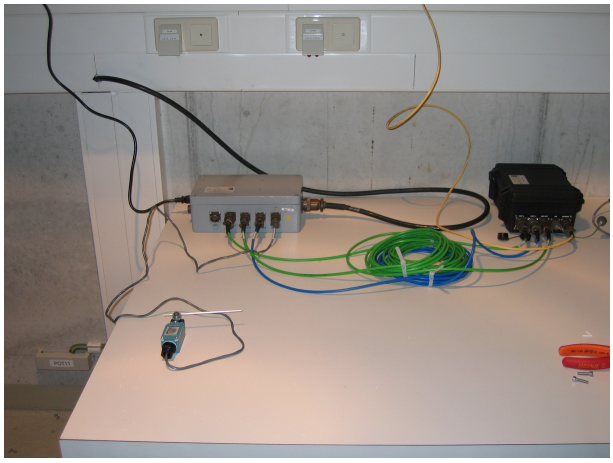
The instrument and DM/AM module were then lowered into the borehole using a winch.

The load-bearing cable was attached to a horizontal stake across the mouth of the borehole.



The head of the borehole was insulated with foam and covered with its lid.

Data cables were run through conduit to the interface box nearby.



For general sensor installation instructions, operational notes and technical information, please see the CMG-3TB manual, supplied separately.

4.1 Testing the borehole installation

You can test the DM/AM module on-site using a laptop computer, by connecting a Güralp serial console cable to one of the two *CONSOLE* connectors of the pit box. The Linux program `minicom`, or `hypertrm` on Microsoft Windows, can be used to communicate over a direct serial link

The borehole installations are of an identical design to those in the vault, except that the DM/AM module contains a modem. The interface box is consequently slightly different, with a 10-pin *NETWORK* connector instead of a 6-pin one. (See Chapter 6, page 35, for details.)

The modem in the interface box is used to communicate with the

downhole equipment. An additional Güralp DCM (Data Communications Module) is supplied which uses the PPP protocol to drive the modem and make the AM in the borehole appear on the local network. This unit is supplied preconfigured to do this. To connect to the network:

- Connect the *NETWORK* connector of the DCM to a network port, and
- connect the *NETWORK* connector of the interface box to the *PORT A* connector of the DCM. The DCM also receives power from the interface box along this cable.

After allowing time for the DCM to set up the modem link, you should be able to connect directly to the *AM's* Web server and configure the AM, digitizer, or sensor.

The DCM also runs a Web server, with a very similar configuration interface to the AM. You should only need to alter any of these settings if the network parameters change; you *cannot* use the DCM to configure the digitizer or send commands to the sensor. Should you need to alter the PPP settings, please see Sreferring also to the Linux documentation for `ppp` and `pppd`.

5 The digitizer module/authentication module unit

This unit combines a Güralp CMG-DM24 digitizer and CMG-AM authentication module in a single sonde casing.

The digitizer-AM units used for the borehole and vault installations differ slightly:

- The digitizer-AM unit for installation in the borehole has an internal long-haul modem, and connects to the interface box with the 10-pin *NETWORK* connector. It is also supplied with a cable strain relief mechanism.

- Digitizer-AM units for installation in the vault do not have this modem. They connect to the interface boxes with 6-pin *NETWORK* connectors.

5.1 Installation

- The top of the digitizer-AM unit has a 26-way mil-spec connector, carrying data, power, tamper inputs, GPS signals and console interfaces for both the digitizer and the AM. This should be joined to the *SENSOR* port on the interface box.

- The bottom of the unit has a 32-way mil-spec plug, carrying signals and auxiliary inputs from the sensor to the digitizer as well as providing the sensor's power supply. This should be joined to the sensor.

The AM is already configured with suitable settings for its internal serial connection to the digitizer. You should not change these settings.

Testing the digitizer

The *DM24 CONSOLE* connector on the interface box allows direct access to the console of the digitizer. To communicate with the digitizer, you will need to set up the parameters of the link. By default, the digitizer is set up to communicate at 38400 baud, with 8 data bits, no parity bit and 1 stop bit.

Once the link is established, pressing `ENTER` will cause the digitiser to reply with its console prompt (`ok` followed by a new line for your commands). You do not have to log out; simply close the connection.

The entire functionality of the digitizer can be controlled by the AM, so you should not need to use its own console during normal operation. If the above steps work correctly, the digitizer may be assumed to be working.

Testing the AM

The *AUTH CONSOLE* connector on the interface box allows you to log in to the AM's Linux operating system, exactly as if you had logged in from another part of the network over a modem connection. You will need a username and password on the AM to do this; an initial `root` password will have been provided to you by Güralp Systems. (If you have not received a password, you should use the distribution's default password `rootme` and change it as soon as you can.)

Once you have logged in, you can check that the AM is receiving data by issuing the command

```
gnblocks 5
```

This will return the number of blocks of GCF data so far received on the digitizer's serial port (normally 5 for a combined unit.)

The AM should now be online and communicating with the array concentrator unit. You can check that the network is up and running by issuing

```
ifconfig
```

This command will list all the active network interfaces (`eth`*`n`* for Ethernet, `ppp`*`n`* for PPP, and `lo` for the local loopback interface), and provide useful information including the number of packets sent and received over each interface. For full details see the Linux manual page for `ifconfig`.

If there is a problem with networking or data transfer, you should check the `/log/messages` file for any errors the AM may have encountered during the start-up process.

Assuming that the AM is working correcly, you can now proceed to configure the pit instrumentation by connecting to it using *ssh* or *https* from anywhere on the local network, including the array concentrator unit itself and PCs on the local end of the Wi-LAN and FreeWave radio connections (if these have been configured).

Accessing the digitizer from the AM

If you need to access the digitizer's command line, you can open a session by running `minicom` from an AM console:

`minicom -n` *`port-number`*

where *`port-number`* is the number of the serial port connected to the digitizer, normally `5`. You can check which port numbers correspond to which devices with the `serialmap` command.

The special `-n` option to `minicom` makes the AM automatically detect the attached digitizer and attempt to bring it into command mode. If it is successful, `minicom` starts normally. When `minicom` exits, the AM returns the digitizer to transmission mode. Standard Linux distributions of `minicom` do not have the `-n` option.

Alternatively, you can send single commands to the digitizer from an AM console using the AM's `gcli` command:

`gcli` *`port-number digitizer-command`*

5.2 Setting up the authentication module

The CMG-AM is at the heart of the system, collecting digitized data from the sensor and producing CD1.1 subframes for transmission. Through the AM in each sensor installation, system operators have complete control over the configuration of that installation.

Under normal circumstances, you will access the AM using its Web page interface, or by connecting to its console using the *ssh* protocol. This interface is fully described in the documentation for the AM.

In addition, you can access the AM's console locally by connecting a suitable serial cable to the *AUTH CONSOLE* port of the interface box. By default, this port operates at 115200 baud, with 8 data bits, no parity bit, and 1 stop bit, and with no flow control.

The configuration database

Most of the AM's configuration settings are held in an internal database called `gcfgdb`. All of these settings are accessible from the Web page interface. They have already been set to suitable values for their role in the array.

If you change any of the settings in the configuration database, be sure to record their previous values, as the AMs are supplied with settings

changed from the default. If you reset an AM to its default configuration, it will no longer work properly as part of the array.

The configuration database can also be updated using command line tools. See the AM documentation for full details.

Authentication

Before any AM installation can start sending CD1.0 or CD1.1 data, it must generate a cryptographic key pair, and request and be assigned a valid certificate. Only with both a key and a certificate can the AM in the borehole produce correctly authenticated data. This procedure can be executed across the network, but must be carried out separately for each borehole.

The program `spyrus` installed on each AM handles the functionality of the on-board cryptographic token. To run this program, you will need to log in to the AM's Linux operating system using *ssh*:

1. From a computer on the local network, use a suitable program to open a *ssh* session with the relevant AM. For example, from a Linux computer:

   ```
   ssh 10.82.0.129
   ```

   where `10.82.0.129` should be replaced with the IP address of the AM on your network.

2. Log in using the username and password you use to access the AM's Web interface. (If you do not use a username and password to access the AM's Web interface, you should obtain one from your network administrator.)

3. Initialise the Spyrus card with the command

   ```
   spyrus zeroize
   ```

   followed by

   ```
   spyrus start
   ```

   You should see information messages coming from the Spyrus card, as follows:

   ```
   Initial state 8(Zeroized)
   info::state: Zeroized
   info::logging into zeroized card
   info::state: Uninitialized
   info::logging into unitialized card
   ```

```
info::loading initialization values
info::state: Initialized
info::changing SSO password
info::state: SSO Initialized
info::logging into the card as SSO
info::loading the trusted certificate
info::state: LAW Initialized
info::logging into the card as SSO
info::setting the user's password
info::state: User Initialized
info::logging into the card as User
info::state: Standby
info::Final state 6(Standby)
```

4. Now log in using

   ```
   spyrus login
   ```

5. Create a text file containing information about the station, sensor, etc., in the following format:

   ```
   organizationName:your-organization
   organizationalUnitName:station-name
   organizationalUnitName:operator-name
   localityName:HPAXX
   commonName:HPA0-02
   commonName:HPA0-01
   ```

   This will be used to request the certificate. You should replace the information after the colons with the correct data for the specific pit installation (available from the NDC or the station administrator.) Check that the field names (before the colons) are correctly capitalized, and do not leave spaces around the colons.

   You can have several `organizationalUnitName` and `commonName` records, in which case the information is entered into the database from bottom to top, so that the most recent entries appear first in the file.

   (To create a text file in Linux, either use a command like `cat > filename` and enter the data directly, ending with `CTRL-D`, or use the text editor `vi`. Using `vi` will allow you to edit the file should you make a mistake in entering the data. For more information, see the Linux manual page for `vi`. Alternatively, you can create text files on the local computer and transfer them to the AM module using `scp` or a similar secure transfer program.)

6. Change into the CD1.1 transmitter's configuration directory using

```
cd /etc/libcd11
```

The following steps create a key pair and certificate request within the token, which need to be placed in this directory for the CD1.1 transmitter to be able to sign outgoing data.

7. Issue the command

```
spyrus newreq -s filename -i 1 -r slot01.req -p slot01.pub
-x
```

where *filename* is the name of the file you created in step 5. This will generate a certificate request in the file `slot01.req` and a public key in the file `slot01.pub`. The private key is kept within the token itself, and cannot be extracted from it. Any attempt to compromise the token will cause it to shut down and become unusable.

8. The file `slot01.req` is a certificate request for the key pair generated. You should send this file by e-mail to the Certification Authority, so that they can generate a valid certificate from it.

9. When you receive the certificate, install it in the `/etc/libcd11` directory as `slot01.crt`. Also create the key ID file `slot01.kid`. containing the key ID as a single decimal number.

10. Now load the certificate into the token using the command

```
spyrus loadcert -c slot01.crt
```

The token will check that the certificate matches its own key pair, and should respond with

```
info::No index specified searching for matching key
info::Key in slot 1 matches certificate
```

If it reports that the key does not match the certificate, you may have attempted to load a certificate valid for the wrong token. Check the certificates you have received, and try again. Otherwise, you may have to generate a new certificate request and re-send to the Certification Authority.

If you issue `spyrus loadcert` without specifying a slot number, as above, any running CD1.1 transmitters will be interrupted, and you will need to restart them.

11. The token is now ready to start signing outgoing CD1.1 subframes. However, you will need to configure the format of these subframes by editing the `/etc/cd11sf.cfg` configuration file. You can do this either directly, or using the Web configuration interface of the AM module.

12. Any further key changes can be handled automatically over AutoDRM. However, occasionally you may want to supersede an existing key, or create a new key for a separate stream.

    Keys are handled by a system of *key buckets*. Each key bucket consists of a list of keys and activation times. Once the activation time for a new key passes, the previous key is superseded, and subsequent subframes are signed by the new key. You can have a different key bucket active for each stream, or even several key buckets for the same stream.

    Key buckets are stored in the files `0.bkt`, `1.bkt`, etc., within the `/etc/keybuckets` directory. Each line in a key bucket file has the format

    key-id:*days-since-epoch*:*seconds-since-day-start*

    where *days-since-epoch* is the number of days elapsed since November 17, 1989. The CD1.1 transmitter scans this file in order, and stops when it finds a key with an activation time in the past (relative to the time-stamp of the data being transmitted.) Thus, to supersede an existing key, you must place the new entry *before* the old one in the file, so that the CD1.1 transmitter will not continue signing subframes with the old key.

    To make the CD1.1 transmitter sign *all* subframes with a new key, even when backfilling, you should add the line

    key-id:0:0

    to the beginning of the file.

13. Restart the CD1.1 transmitter with the command `killall -HUP cd11sf`. (Using the `-HUP` option makes the command send a hangup signal to the CD1.1 transmitter rather than killing it outright.)

CD1.0 and CD1.1 transfers

The AMs do not transmit CD1.1 data direct to data consumers. Instead,

they generate authenticated subframes and transfer these to the data server using HTTP. These transfers are initiated by the data server (see section 7.2, page 53.) You do not need to set any of the options under *datatransfer.cd1* or *datatransfer.cd1_1*, as these options are intended for direct transmission only.

The AM generates CD1.1 subframes, which have some flexibility in their format. A configuration file, `/etc/cd11sf.cfg`, is provided which allows you to specify the exact format to be used in outgoing CD1.1 subframes.

The Web interface also includes a form which enables you to edit the file and its attributes directly.

The Web interface does not check that the content of the files will be understood. You should ensure that the file is valid before committing any changes.

The format of the file is as follows:

```
HPA1Z2:HPA1CD:HP:HPA1:BHZ:/data/HPA1.:141
HPA1N2:HPA1CD:HP:HPA1:BHN:/data/HPA1.:141
HPA1E2:HPA1CD:HP:HPA1:BHE:/data/HPA1.:141
```

Each line represents an incoming stream from the digitizer; the fields

are separated by colons, and are in turn:

- the digitizer's data stream ID;
- the digitizer's status stream ID;
- the CD1.1 location code for the array;
- the CD1.1 site code for the instrument;
- the CD1.1 channel name;
- a prefix for the location to store the files in the file tree of the AM; and
- the key-bucket code, which tells the AM which key to use for that particular stream.

To produce unauthenticated CD1.1 data, you should use a keybucket code of `-1`

The prefix is used to determine where to place generated CD1.1 subframes. For example, a prefix of `/data/HPA1`. will produce files in the `/data/` directory beginning with `HPA1`. and followed by a unique timestamp. Several streams may produce files using the same prefix; indeed, this is recommended.

When you have finished editing the file, clicking **Save changes** will write the changes to disk. The changes will not take effect, however, until the CD1.1 service is restarted. You can restart all running CD1.1 services by clicking **Restart CD1.1**.

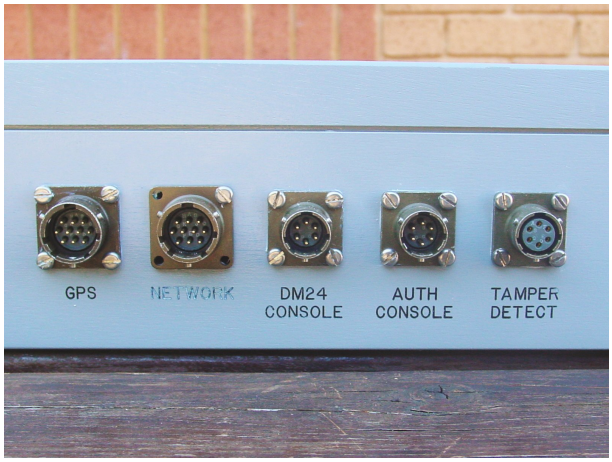
CD1.0 subframes are generated in exactly the same way, taking their configuration from the file `/etc/cd10sf.cfg`. The configuration options are the same.

6 The interface box

Each sensor installation has an interface box, which takes the single data cable from the digitizer-AM unit and splits it by function using standard Güralp connectors. The interface box may or may not include a built-in modem: boxes with 10-pin *NETWORK* connectors have a modem installed, whilst those with 6-pin *NETWORK* connectors do not.

6.1 Connections

The front face of the box has the following connectors:



- The *GPS* connector is a 10-pin mil-spec plug, and should be connected to a Güralp GPS module, or a GPS repeater system using an appropriate cable.

- The *NETWORK* connector is a mil-spec plug with *either* 6 or 10 pins.

In the 6-pin variant, the *NETWORK* connector is a standard Ethernet port, passed through directly from the AM, and should be connected to your local area network.

In the 10-pin variant, the *NETWORK* connector is the output from a built-in modem. This type of interface box is designed for use in installations where the digitizer-AM unit has a built-in modem (normally boreholes.) The *NETWORK* connector on these boxes is in fact a serial port; to connect this to your local area network, you will need a separate DCM or serial-to-Ethernet converter which can be configured to run PPP or SLIP (see below.)

- The *DM24 CONSOLE* connector is a 6-pin mil-spec plug, which can be connected to a PC's serial port for emergency direct access to the digitizer console.

- The *AUTH CONSOLE* connector is a 6-pin mil-spec plug, which can be connected to a PC's serial port for direct access to the AM's command line. You may need to use this connector to set up the AM before its Web server is up and running.

- The *TAMPER DETECT* connector is a 6-way mil-spec socket, which can be connected to up to 5 additional tamper switches (*Tamper 0 – 4*). These switches will be reported on the AM's tamper evidencing page. In addition, *Tamper 0* sets the *VAULT DOOR OPEN* flag in outgoing CD1.1 subframes.

  There is also a microswitch inside the interface box, which sets the *Tamper 5* line in the AM and the *EQUIPMENT HOUSING OPEN* flag in outgoing CD1.1 subframes.

On the left side of the interface box are two further connectors:

i

The 3-pin standard *MAINS* plug should be connected to a mains (outlet) power supply at 100 – 240 V AC. This will power the sensor and digitizer-AM module, but not the hole lock unit of a borehole sensor. This plug should be used for the borehole installation.

The *POWER AND HOLELOCK* connector is a 10-pin mil-spec plug, which may be connected to a 10 – 36 V DC power supply as an alternative to the *MAINS* plug for the vault installations. This connector should also be used to plug in a Holelock Control Unit to operate the hole lock unit of a borehole sensor. This arrangement makes sure the hole lock and sensor power circuits are completely isolated from each other.

On the right side of the interface box is a single 36-pin *SENSOR* plug, which should be connected to the signal cable from the digitizer-AM module.

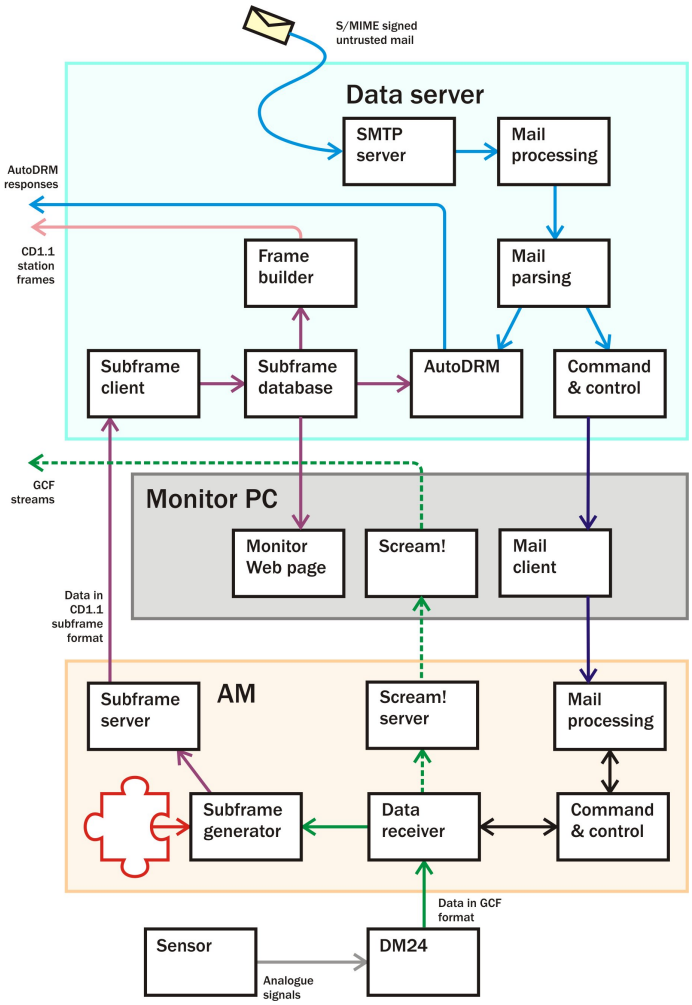
Interface boxes with 10-pin *NETWORK* connectors expose a direct serial link to the AM in the sensor installation *via* a modem link. You cannot connect these boxes directly to your network. Instead, an additional Güralp DCM (Data Communications Module) can be used as an interface between the network and serial connections, and use the PPP protocol between this DCM and the AM in the installation.

7 The data centre

The data centre is responsible for collating CD1.1 subframes from the various instruments and forming full CD1.1 station frames from them, which it sends to the NDC or other receivers as required using the protocol defined in the standard. This core functionality is performed by the dedicated data server, which runs the Linux operating system and is intended to run full-time.

A second PC is provided for day-to-day use by the operator, which includes e-mail and Web clients for use in administering the system. Using a separate PC for monitoring ensures that the resources of the server can be prioritized for CD1.1 transmission.

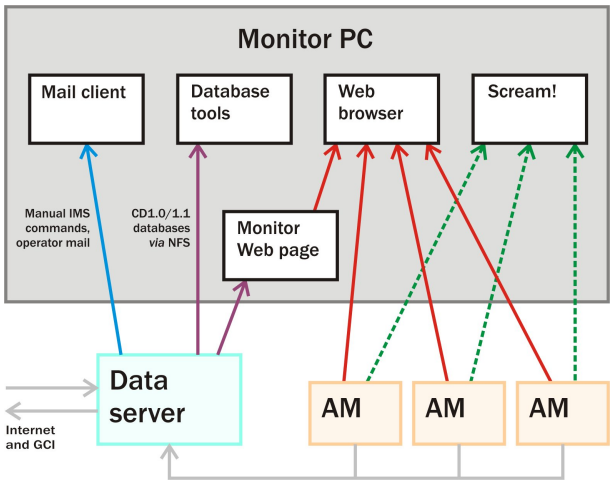
Each machine forming part of the array runs a number of software processes to carry out its designated task. These processes are linked together according to the diagram below. Connections between machines shown on this diagram do not represent direct physical links; instead, each connection represents a communication pathway over the physical network described in the previous chapters.



7.1 The monitor PC

The monitor PC is the main interface to the system for the station operator. It communicates with the data server and AM units using standard protocols. It is supplied running a standard distribution of the Linux operating system, using open-source applications where possible.

The diagram below shows the system software arrangement from the point of view of the monitor PC.



The monitor PC has two network interfaces:

- One interface is connected to the data server and instruments *via* an Ethernet switch. This interface is used for all incoming data.

- A second interface is provided for use by the local institution, *e.g.* for additional data transmission in (unauthenticated) GCF format. The provided Scream! software can be used for this purpose, as well as (optionally) for monitoring data flow.

The following sections describe the main applications which run on the monitro PC:

Mail client

The monitor PC is shipped with the Mozilla Thunderbird mail client. This forms part of the AutoDRM mechanism as depicted above.

1. A S/MIME signed message in IMS2.0 format is delivered to the AutoDRM in the data server.

2. The data server checks the S/MIME signature on the message, and tests it for integrity and validity.

3. If the data server is so configured (see Section 7.2, page 53) IMS2.0 commands relating to key and certificate management are relayed to each AM unit for automatic action there.

4. If the IMS2.0 command requires action by the operator (*e.g.* calibration), is unrecognised, or if you have disabled automatic command handling, the AutoDRM message is delivered to a local mailbox. The operator should check this mailbox using the mail client and action any AutoDRM requests they find.

The mail client can also be used to handle mail direct to the operator using existing POP3 or IMAP servers. See its own documentation for details of how to do this. Note that you should *only* use the monitor PC for this purpose, and never the data server.

Web browser

The monitor PC uses the Mozilla Web browser. This forms the main means by which station operators configure and monitor the station.

- The monitor PC itself runs a compact Web server, which graphs information on the status and integrity of the data held on the data server.

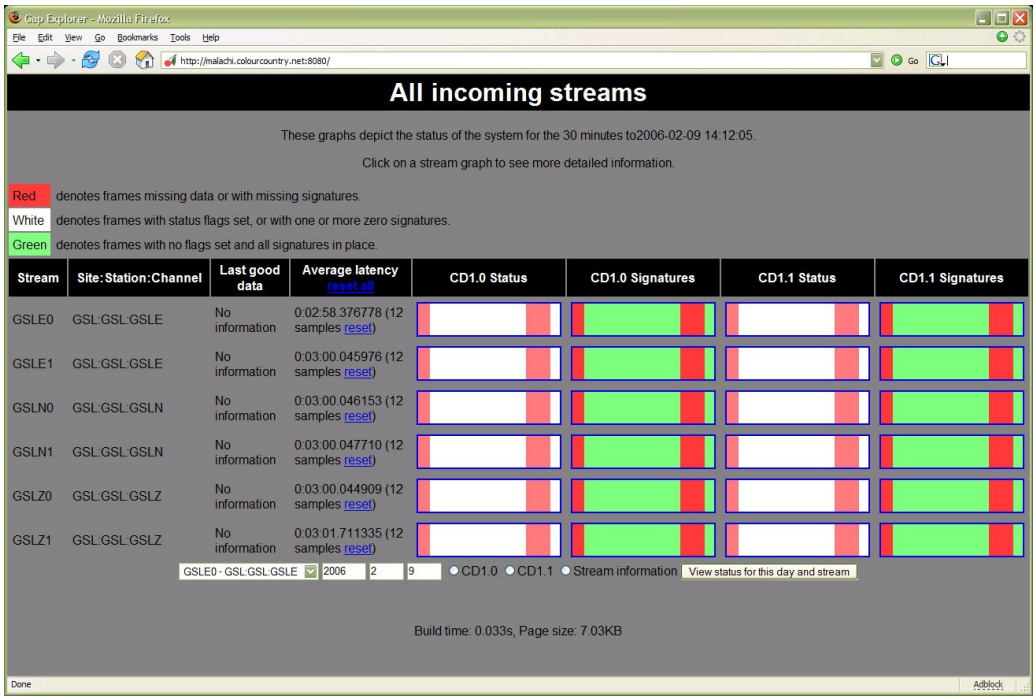
- Each AM has its own Web page, which can be used to configure both itself and the attached digitizer and instrument.

Monitoring data flow

The monitor PC's Web server connects to the subframe database and presents information about its contents graphically. To view this information, enter the URL

```
http://127.0.0.1/
```

This will bring up a page showing the integrity of recent data produced at the station. The page will automatically reload every minute.



The table gives an overview of the system status in the last 30 minutes. Each entry in the table corresponds to one of the streams currently being received at the data server. The columns are:

Stream : The internal (GCF) stream name for this stream.

Site:Station:Channel : The CD1.0/1.1 stream designations for this stream.

Last good data : The time stamp on the latest data from any of the instruments that does not report any problem (*i.e.* the data exists, and no CD1.1 status flags are set.)

Average latency : The difference between the time data was requested (as measured by the monitor PC's own clock) and the timestamp of the latest data returned. Because CD1.0/1.1 subframes are emitted every 10 seconds, you should expect this value to be in the region 5 – 10 seconds under normal operation.

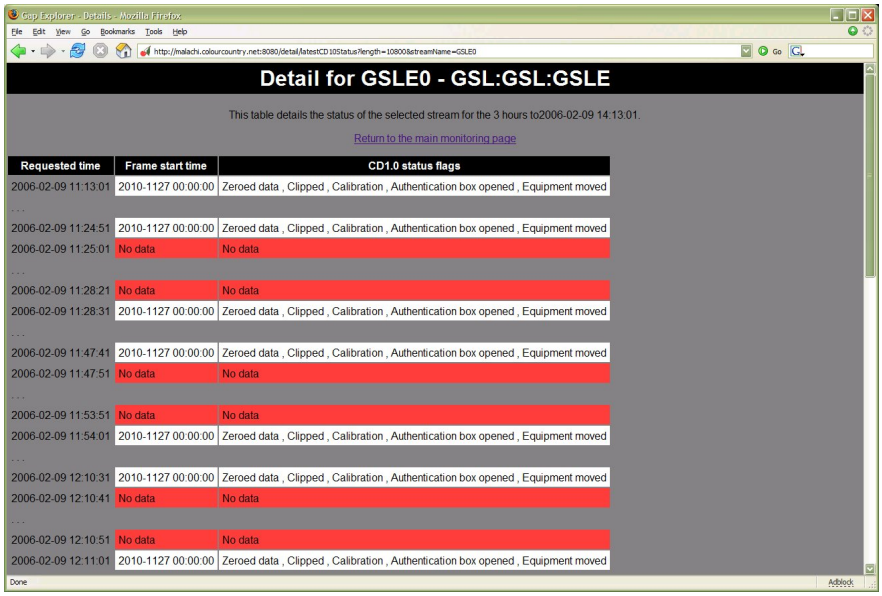
The Web page measures the latency of every stream each time it is refreshed, and displays a cumulative average of these measurements. The legend *4 samples* would denote that 4 measurements have been taken of this stream to give the average latency displayed. To make the page forget the current running average for a stream, click the **Reset** link in the table entry. To make it forget the average for all streams, click the **Reset** link in the heading.

CD1.0 Status : This graph shows whether any CD1.0 status flags have been set in the last 30 minutes. The current time is on the right of the picture; the left edge of the picture corresponds to a time 30 minutes ago. Each 1-pixel vertical stripe corresponds to one subframe.

A green stripe denotes that CD1.0 data for this time period is present, and no status flags are set. A white stripe denotes that one or more status flags is set. A red stripe denotes that data for this time period is unavailable.

Because of the normal latency between the instruments and the data centre, a station in normal operation will show a green image with one or two red pixels at the right edge (corresponding to data that has not yet arrived.)

To view details of the CD1.0 status flags, click on the image. A new page is displayed:

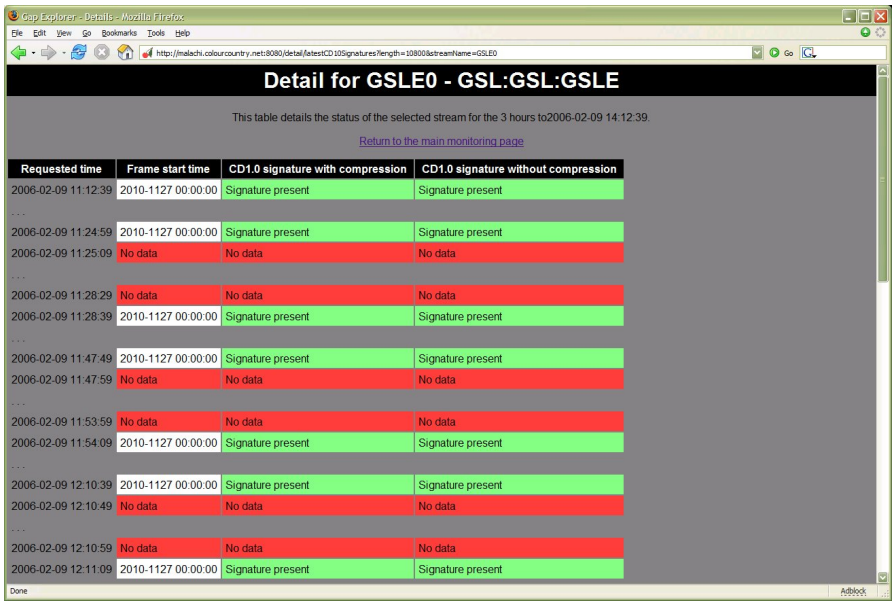


This page describes which CD1.0 status flags have been set over the last 3 hours. Each subframe corresponds to a single row of the table. If many adjacent subframes have the same status flags set, they are collapsed together to save space.

Click **Return to the main monitoring page** to go back.

CD1.0 Signatures : This image is like the *CD1.0 Status* image, but it describes the current state of the signatures on the data. Green pixels denote that all signatures are present; white pixels, that one or more signatures is zero or missing, and red pixels that the data is missing entirely (as for the *Status*.)

Click this image for a detailed 3-hour summary of the signature status. Each of the CD1.0 signature options is reported separately.



CD1.1 Status : This is identical to *CD1.0 Status*, except that it monitors the CD1.1 status flags. Click on this image for a 3-hour detailed view.

CD1.1 Signatures : This is identical to *CD1.0 Signatures,* except that it monitors the 3 CD1.1 signature types. Click on this image for a 3-hour detailed view.

At the bottom of the mian monitoring page is a form allowing you to view the details of any stream in the data server's database by date. To retrieve this historical data:

1. Select the stream you wish to view from the drop-down list.

2. Enter the date of interest into the three text boxes, year, month and day from left to right.

3. Select *CD1.0* to view CD1.0 status and signature information, *CD1.1* for CD1.1 status and signature information, or *Stream information* to view stream text fields.

4. Click **OK**. The monitor PC will then request the data from the server. Because an entire day of data will be displayed, you should expect this to take several seconds.

If a stream or instrument is not appearing in the subframe database, you will see red pixels in the *Status* and *Signatures* images for that stream or instrument. You can troubleshoot the instrument by

connecting to the Web server of the AM in the same pit and running the *Data Viewer* applet (under *Actions*). Alternatively, you can set up each AM to run a Scream! server, and run a copy of Scream! on the monitor PC to view the data.

Configuring AMs

To connect to an AM, enter the URL

`https://10.82.0.123/`

replacing 10.82.0.123 with the IP address of the AM you want to connect to.

For full details of the services provided by an AM's Web server, see the AM user manual.

To edit files on the data server, you need to log in using `ssh` and issue commands as an authorised user. Sessions on the data server do not have access to the files on the monitor PC. For example, to edit the `/etc/inittab` file on the data server (user input is marked in bold):

```
[monitorpc ~] ssh -l root dataserver.local
Password: ********
[dataserver ~] vi /etc/inittab
```

The data server will need to be configured before it is commissioned, by editing certain configuration files as described below. The operator of the monitor PC will not need to edit files on the data server during the normal operation of the system.

Database tools

The monitor PC has a suite of command-line tools for tracing the origin of data frames sent by the CD1.1 transmitters. Two databases are stored on the data server for each transmitter:

- a *frame database* which relates frame numbers to frame offsets (streams) and time stamps, and
- a *data database* which locates the data for a given frame offset and time stamp, if it is present.

The frame and data databases record all frames which have been sent by the data server. After a sufficiently long period, the actual data forming these frames may be overwritten or deleted. However, these databases remain.

The databases are exported to the monitor PC for reading only, at the mount point `/data`. They may be inspected using the commands `framedbdump` and `datadbdump`. The frame and data databases for each running CD1.1 transmitter are stored within the `cd11xmit` working directory. Paths described below are relative to the working directory of the CD1.1 transmitter.

datadbdump

The `datadbdump` command provides information on the data contained in transmitted frames, including the location of the data on the PC's hard disk. The syntax is

`datadbdump` *`frame-offset`* *`day-number`* (*`record`*)

where *`frame-offset`* is the frame offset of the stream (as related to data streams by the `config/framepos` file; see above), *`day-number`* is the date of the data, expressed as a number of days elapsed since November 17, 1989, and *`record`* is the record number from that day to use. If *`record`* is missing, `datadbdump` will display all the records for that day.

The dump will be output as a series of records formatted similar to this:

```
datadb: F 2/D 5347/    0 (Thu Jul  8 00:00:00 2004) [Sent]
  in /data/lumps/SPA0.  5347 offset 3112
```

The fields are:

- the word `datadb:`,
- the frame offset of this stream (`F 2`),
- the time-stamp of the data, expressed as a number of days elapsed since November 17, 1989 (`D 5347`),
- the record number (`0`),
- the time-stamp of the data in human-readable format,
- the status of the data (`Sent`),
- the location of the data file, shown as a prefix and day number (`/data/lumps/SPA0.5347`, formed from the prefix `/data/lumps/SPA0.` and the day `5347`),

- the offset of the data within the file, in bytes.

framedbdump

You can use the `framedbdump` command to inspect frames that have been transmitted by changing to the `cd11xmit` working directory (*e.g.* `/var/cd11xmit_1`) and issuing

```
framedbdump frame-number (frame-numbers...)
```

If the frame exists, a dump will be output in a format similar to this:

```
Frame 374000:
  day  5346 recno  7838 (Wed Jul  7 21:46:20 2004)
  last sent Wed Jul  7 21:49:56 2004
  { 0 1 2 3 4 5 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29
 30 31 32 33 34 35 }
    datadb: F 0/D 5346/ 7838 (Wed Jul  7 21:46:20 2004) [Sent]
      in /data/lumps/SPA0.  5346 offset 9738528
      ...
```

At the top of the dump are reported:

- the frame number (`Frame 374000`),
- the time-stamp of the frame, expressed as a number of days elapsed since November 17, 1989 (`day 5346`),
- the record number (`recno 7838`),
- the time-stamp of the frame in human-readable format,
- the date this frame was last sent to an NDC or other CD1.1 receiver, in human-readable format, and
- the frame offsets which are present in this frame (as related to data streams by the `config/framepos` file; see above).

Following this is a list of all the data making up the frame, in the same format as returned by `datadbdump`.

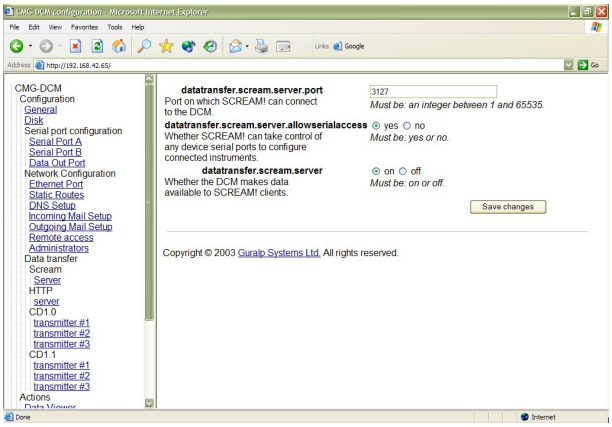
Scream!

The monitor PC is provided with Güralp Systems' Scream! software for displaying and recording data from the AMs.

Before you can use Scream! to view data, you will need to instruct each AM to run a Scream! server. To do this:

1. Use the Web browser on the monitor PC to access the Web site of the AM.

2. Under *Data transfer* → *Scream*, click **Server**.



3. Set the options *datatransfer.scream.server* to *on* and *datatransfer.scream.server.allowserialaccess* to *yes*. Make a note of the value of *datatransfer.scream.server.port*, or change it to a different port number.

4. Click **Save changes**.

The AMs will now provide data to Scream! when requested. For security, you may want to configure the firewall between the station and the Internet or GCI to block connections on the port you have chosen, so that only the monitor PC can connect in this way.

1. Run Scream! with the command `scream`

2. From the menu in the main window, select *Windows* → *Network Control*. Switch to the *My Client* tab if necessary.

3. Right-click beneath *Servers* in the top panel. Select **Add UDP Server...**.

4. In the dialogue box, type in the IP address of the AM and the port you chose for *datatransfer.scream.server.port*, above, *e.g.*

   ```
   10.82.0.123:8888
   ```

   Press `ENTER` to register the AM in Scream!'s server list.

5. Right-click on the entry in the server list and select **GCFPING**. You should see an alert box with the message *Ping acknowledged*.

   If you do not, there is a problem with your network settings.

Check the IP address and port, and that the connection to the AMs is operational.

6. Right-click on the entry in the server list and select **GCFSEND:B**.

7. Repeat steps 3 – 6 for the other two AMs.

Data should now start appearing in Scream!'s main window.

For full information on viewing and recording data in Scream!, see the supplied Scream! manual.

Running several Scream! instances simultaneously

In certain circumstances you may want to run two or more instances of Scream! simultaneously on the monitor PC, with different configurations. For example:

- A single instance of Scream! can only record in one data format. If you want to record in several formats simultaneously, you will need to be running as many instances of Scream! as there are data formats to be recorded.

- When recording data, Scream! will use its internal stream buffer to sort out-of-sequence data as far as possible. This means that if a gap occurs, Scream! will stop recording data until *either* the gap is filled, or the stream buffer is filled with new data. Only then will it write out the data. If you want to use a large stream buffer for monitoring, but expect there to be gaps or out-of-sequence data, you will need to use one instance of Scream! (with a small stream buffer) for recording, and a separate instance (with a larger buffer) for monitoring.

- You may want to run one instance of Scream! constantly in the background, and use a second instance to monitor the incoming signals from time to time.

Using separate configuration files

By default, Scream! uses the configuration file `scream.ini` to set itself up automatically when restarted. If you want to run two Scream! instances with different options, you will need to create a separate configuration file for each one. You can run Scream! with an alternative configuration file with the command-line option `/i`. For example, to make Scream! use the defaults in the file `/root/recorder.ini`, you would issue the command

```
scream /i:/root/recorder.ini
```

If you make changes to the configuration of this Scream! instance and quit, it will write out the new configuration to this file instead of to `scream.ini`.

By default, a Scream! server will not pass on data which it received over the network, but only from the local serial ports. This is done to prevent infinite loops occurring where Scream! servers simply bounce data from one to another, using up all the available bandwidth.

However, if you are using two Scream! instances on the same machine, you will normally want the second instance to see *all* the data sources recorded by the first one. To do this, you will need to launch the *server* with the command line option `/nettx`, *e.g.*

```
scream /nettx /i:/root/server.ini
```

To prevent loops from being created accidentally, you should omit this option from the client instance.

Communicating between Scream! instances

In most cases where you are running several Scream! instances, you will want them to be able to pass data between each other. You can do this using your computer's *loopback* network interface, which uses IP addresses in the range 127.0.0.1 – 127.255.255.255. The steps below will let you use two Scream! instances to process and view data coming in over the computer's serial interface.

If you want to make this setup launch automatically when you log in, you should make sure that the various instances of Scream! are launched in the right order.

1. Use the command

   ```
   scream /nettx /i:/root/server.ini
   ```

   to launch the instance of Scream! which is currently connected to the instrumentation. This Scream! is going to receive data, and perhaps record it in real time. The `/nettx` option is necessary because we want Scream! to retransmit data which it received over the network.

   If you want any serial ports to be available to other instances of Scream!, ensure that *(none)* is selected as the baud rate for these ports (in the *Com Ports* tab of the *Setup* window) and save your changes before starting any more Scream!s.

2. Choose *Windows* → *Network Control* from the main window.

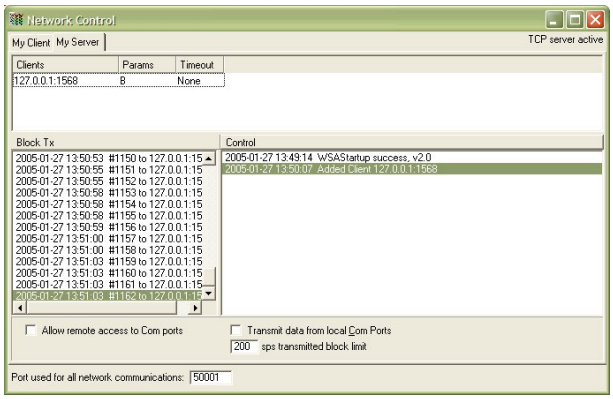
3. Click on the *My Server* tab.

4. Right-click beneath *Clients* in the upper table, and select **Add...**.

5. Enter the IP address and port `127.255.255.255:1568`.

   The IP address `127.255.255.255` means "broadcast to all local clients listening on this port". If you are running more than 2 Scream! instances, you must use this IP address to receive all the data. If you use the more common `127.0.0.1` address, only one Scream! client will be able to connect.

   You may use any port number above 1024 as long as it is not being used by another application. By default, Scream! listens on port 1567 for data from your seismic network. This is what the first instance of Scream! will be doing. The second instance of Scream! will also need to receive data, but *only* from the first one, so we must use a *different* port number for it: here, 1568.



6. Check the *Transmit data from local Com ports* box.

7. Launch a new instance of Scream! with a different configuration file:

   ```
   scream /i:/root/monitor.ini
   ```

8. Ope n the *Network Control* window on the new instance of Scream!, and click on the *My Client* tab.

9. In the *Port used for all network communications* box, type `1568`, or the port number you used in step 5.

10. Check the *Receive UDP Data* box. The local loopback should

now be active, and if data is coming in, you should see it appearing.

11. Configure the second instance of Scream! as required.

Additional tools

`ssh` allows station operators to connect securely to other hosts on the network, including the AMs. To log in to one of the AMs, you will need to use the same username and password that you use to access the Web interface. You will be given a Linux command prompt. At this prompt, you can use AM terminal commands, *e.g.* `gcli` and `minicom` to communicate with digitizers, `gnblocks` to query incoming data, and `gcfgdbget` and `gcfgdbset` to alter the AM configuration options. See the AM documentation for full details of the commands you can issue.

`updown` monitors the health of the system by sending ICMP ping packets to each of the AM units and listening for replies. The results are output in visual form For each unit, a green box denotes that the unit is responding to ICMP ping requests, and a red box indicates that the unit has stopped responding.

Once running, the `updown` program polls all units every two seconds. However, in order to conserve bandwidth the program does not immediately poll the whole system on strtup. You should allow 20 seconds for the procedure to complete.

`smartd` runs on the data server as well as on the monitor PC, and monitors the local hard disks using their built-in Self-Monitoring Analysis and Reporting Technology. It can detect when a hard disk is about to fail, and e-mail alerts to a specified address. `smartd` is run from `/etc/inittab` on each machine, and you can change the options used (including the destination e-mail address for alerts) in `/etc/smartd.conf`.

`THE DOCTOR` runs on the data server, and monitors the CD1.1 transmitter process. If the process exits abnormally, `THE DOCTOR` automatically saves a traceback in `/blackbox/`*`current-unix-time`*`/where`. The traceback details any error messages given, and the name of the function the CD1.1 transmitter was in at the time. If a serious error occurred, a core dump will be saved in `/blackbox/`*`current-unix-time`*`/core.`*`process-id`*. A new CD1.1 transmitter will then be spawned (by `init`) to replace the defunct one.

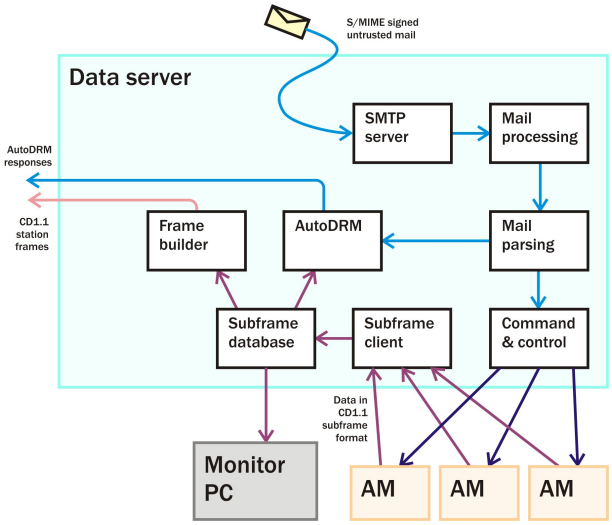
`ncad` listens on port 911 of both the data server and the monitor PC, and provides *ssh* access direct to the Linux console. You will need to log in as `root`, or a user with full access to the console devices, to do this. Using `ncad` you can view boot-up messages as they are produced, and if the boot sequence is interrupted (for example, for filesystem checking) you can perform the required administration remotely. All characters except `<Ctrl-A>` are sent directly to the console; `<Ctrl-A>` brings up a short command menu where you can change virtual terminals, quit the program, or spawn a shell. Because `ncad` displays a reproduction of the actual Linux console, you must have a display at least as large (in characters) as the actual PC monitors; `ncad` will tell you if this is the case.

The Python application which displays the Web monitor interface, and runs the monitor PC's compact web server, is located in the directory `/software/apps/python-gap-explorer`. It is automatically launched by the file `/etc/inittab` .

7.2 The data server

The data server PC runs the same Linux distribution as the monitor PC, but its software is specialized for its role in the array. It runs subframe transfer and collation services, a CD1.1 transmitter, and protected e-mail services. It also exposes a secure shell (`ssh`) interface. All other access is prevented by the firewall.

The software layout from the point of view of the data server is shown in the diagram below.



Receiving CD1.1 subframes

Each Linux PC runs a CD1.1 transmitter which it uses to relay streams which it fetches from the pit installations. The transfer is performed by a service named `lxclient`, which is run on startup. The `/etc/inittab` file is responsible for starting the `lxclient` services.

The format of the `lxclient` command is:

`lxclient` *`url-of-transfer-script`* *`prefix`* [*`prefixes...`*]

The *`url-of-transfer-script`* parameter is a URL from which the CD1.1 transmitter can fetch ready-formatted CD1.1 subframes, normally beginning *http://* or *https://*. AMs in the pit installations provide two *cgi* programs for access to these subframes, named `lxserver` and `lxstream`, which you can access by supplying a username and password as part of the URL. The difference between `lxstream` and `lxserver` is that `lxserver` will read all waiting CD1.1 subframes and return, whilst `lxstream` will continuously transfer incoming data to the CD1.1 transmission area.

Thus, in the standard setup the `lxclient` command takes the form

`lxclient http://root:rootme@`*`my-borehole-am`*`/cgi-bin/lxstream` *`prefix`* [*`prefixes...`*]

using the username `root` with the AMs' distribution default password of `rootme`. The `/etc/inittab` file will contain at least one `lxclient` entry for each of the borehole AMs.

Any valid user of an AM is authorized to retrieve CD1.1 subframes; when you change the `root` password on the AM, or if you want to change the username under which the transfer takes place, you must alter the `/etc/inittab` file accordingly.

One or several groups of streams may be transferred simultaneously by `lxclient` by specifying the relevant prefixes on the command line. The *`prefix`*(es) are those used to identify stream groups in the CD1.1 subframe configuration file (on the AM), `/etc/cd11sf.cfg`; see Section , page , for details.) Prefixes end in a full stop, and correspond to a number of files, where each file's name begins with the prefix and continues with a time stamp. Several `lxclient` processes may work on the same group of streams without affecting the data server's ability to format complete CD1.1 frames.

By default, `lxclient` will place retrieved CD1.1 subframes in the directory from which it is called. You can provide a different directory

if you wish using the -c *directory-name* option. The initial setup of the PCs is to have lxclient place incoming CD1.1 subframes in the directory /export/data/cd11_lumps/.

You should take care when altering or removing data files created by lxclient, in particular files which are incomplete. lxclient is not designed to handle general synchronization between machines, and only supports data being placed in new files or appended directly to existing files. Other parts of the armoured database will not be corrupted if this happens, but you may lose data.

Transmitting CD1.1 frames

Once CD1.1 subframes are waiting in the CD1.1 transmission area, the data server is ready to format them into complete frames. The cd11xmit process handles assembling the frames and submitting them to the NDC or other CD1.1 receiver. cd11xmit is run automatically from /etc/inittab, initially with the command line

```
cd11xmit -d /var/cd11xmit_1 < /dev/tty6 > /dev/tty6 2>&1
```

The -d option tells cd11xmit to use the working directory /var/cd11xmit_1. The remainder of the command line redirects output to virtual terminal 6, so that you can monitor the frames being sent by pressing <Ctrl>-<Alt>-<F6>. Press <Alt>-<F7> to return to X.

You can have several CD1.1 transmitters running simultaneously, operating out of different working directories, by adding more cd11xmit commands to the /etc/inittab file. You will need to create the new working directory yourself, which should contain:

- a config/ directory with framepos and config files as detailed below;
- an empty framedb/ directory for the frame database;
- an empty datadb/ directory for the data database.

The framepos file

The format of the CD1.1 frames is determined by the configuration file config/framepos within the cd11xmit working directory, which has the following format:

```
0:HP:HPA0:BHZ:HPA0Z2
1:HP:HPA0:BHN:HPA0N2
2:HP:HPA0:BHE:HPA0E2
```

```
3:HP:HPA1:BHZ:HPA1Z2
4:HP:HPA1:BHN:HPA1N2
5:HP:HPA1:BHE:HPA1E2
...
```

Each line in the file represents a stream for inclusion in each formatted CD1.1 frame, in the order in which they will appear in the frame. The fields, which are separated by colons, are in turn:

- the position of the stream in the frame, starting from 0;
- the CD1.1 location code for the array;
- the CD1.1 site code for the borehole;
- the CD1.1 channel name;
- the borehole digitizer's data stream ID.

The config file

The destination for CD1.1 data is specified in the configuration file `config/config` within the working directory. This file contains a number of lines of the form *`variable value`* (space separated). You can comment out any line by prefixing it with a `#` symbol.

```
name HPAXX
creator HPAXX
type NDC

dest IDC
destdata 0

source /data/lumps/HPA0.
source /data/lumps/HPA1.
...
source /data/lumps/HPX3.

receiverip 250.251.252.253
receiverport 10000
```

The variables you can use are:

name : The 5-character name of this station.

creator : The 5-character name of the station creating the CD1.1 frames (normally the same as *name*.)

type : The station type of this station (`IDC`, `IMS` or `NDC`.)

dest : The frame destination to use for connection requests, option

requests and alert frames (`0` if indeterminate.)

destdata : The frame destination to use for data frames (`0` if indeterminate.)

source : A prefix for a data stream to send to the specified destination. There will normally be several of these. The prefixes are those used to identify stream groups in the CD1.1 subframe configuration file (on the AM), and also those used by the `lxclient` program described above.

receiverip : The IP address of the receiving station.

receiverport : The port number to use at the receiving station.

IMS2.0 commands and AutoDRM

The data server runs a full-featured AutoDRM client which can be used to request data. Additionally, station operators can send IMS2.0 commands over the same interface to configure the system. The accepted commands are:

`CALIBRATE_START` – Request a calibration experiment. This experiment must be performed manually by a station operator. These messages are forwarded to the local mail spool for scheduling and later action.

`UPDATE_CRL` – Update the Certificate Revocation List used to authenticate users issuing IMS2.0 commands.

`GENERATE_KEYPAIR` – Instruct the station to generate a new key pair for signing outgoing data.

`START_KEYPAIR` – Instruct the station to start using the previously-generated key pair.

Authentication

IMS2.0 commands are encapsulated within e-mail messages authenticated using S/MIME. Before a station operator can use IMS2.0 to control the station, they will need to be added to the list of users permitted to execute IMS2.0 commands. To add a user:

1. Log in to the data server as `root`.

2. Ensure that the directory `/export/home/imsparser/certificates/` exists on the data server.

3. Create a keyring file named after the user, which contains the certificate(s) currently in use by the new user. The file should have no extension.

4. Ensure that the directory `/export/home/imsparser/acl/` exists on the data server.

5. Create an ACL file named after the user, which contains the list of IMS2.0 commands permitted to this user, *e.g.*

   ```
   CALIBRATE_START
   ```

   This file must have the same name as the one in the `/export/home/imsparser/certificates/` directory.

   The user can now sign messages using the private key corresponding to the certificate stored in `certificates/`, and issue any command listed in `acl/`.

8 Connector pinouts

8.1 Vault sensor output port

Models with a 26-pin mil-spec plug (02E-16-26P) have the following pin assignments.

Pin	Function
A	Velocity +ve, vertical channel
B	Velocity –ve, vertical channel
C	Velocity +ve, N/S channel
D	Velocity –ve, N/S channel
E	Velocity +ve, E/W channel
F	Velocity –ve, E/W channel
G	Mass position, vertical channel
J	Mass position, N/S channel
K	*BUSY* LED
L	Mass position, E/W channel
M	Power –ve (not used in sensors with internal DC-DC converters)
N	Signal ground
P	Calibration signal (all channels)
R	Calibration enable, vertical channel
S	Calibration enable, N/S channel
T	Calibration enable, E/W channel
U	Centre
W	Unlock
X	Lock
Y	Logic ground
b	Power ground
c	Power +ve

8.2 Borehole sensor output port

This connector is a 32-way mil-spec waterproof plug (02E-19-32P).

Pin	Function
A	Hole lock motor
B	Hole lock motor return
C	Inclinometer power
D	+ 12 to 24 V DC power
E	0 V power
F	Mass position, vertical channel
G	Mass position, N/S channel
H	Mass position, E/W channel
J	Velocity +ve, vertical channel
K	Velocity –ve, vertical channel
L	Velocity +ve, N/S channel
M	Velocity –ve, N/S channel
N	Velocity +ve, E/W channel
P	Velocity –ve, E/W channel
R	Calibration signal (all channels)
S	Calibration enable (all channels)
T	Y tilt
U	X tilt
W	*BUSY* LED
V	Centre
X	Direct
Z	Unlock
Y	Lock
a	Guide switch LED
b	*Acc*/*Vel* mode switch**
e	Signal ground
f	Control signal ground
j	Case ground

**The *Acc/Vel* mode switch pin is used by the automatic centring process to switch the instrument temporarily into a mode where its response at normal frequencies is flat to acceleration, and thus in linear proportion to the mass position. The surface control unit returns the sensor to its operational mode once the centring process is completed.

8.3 DM/AM module output port

This is a standard 32-pin mil-spec plug (02E-18-32P).

Pin	Function
D	Power +10 to +30 V
E	Power 0 V
J	Data transmit +ve
K	Data transmit –ve
L	Data receive +ve
M	Data receive –ve
R	AM console receive
S	AM console transmit
T	AM console ground
U	Tamper switch 5
V	GPS 1pps signal
W	GPS RS232 transmit
X	GPS RS232 receive
Y	GPS RS232 ground
Z	DM console transmit
a	DM console receive
b	DM console ground
c	Tamper switch 0
d	Tamper switch 1
e	Data ground
f	Tamper switch ground
g	Tamper switch 2
h	Tamper switch 3
j	Tamper switch 4

8.4 GPS port

This is a standard 10-pin mil-spec plug (02E-12-10P).

Pin	Function
A	Power 0 V
B	Power +12 V
C	PPS
G	RS232 ground
J	RS232 transmit to GPS
K	RS232 receive from GPS

It is connected to the Güralp GPS receiver by a pin-to-pin cable.

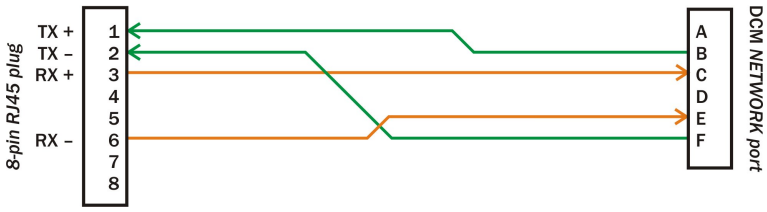
The cable provided also connects the unused pins, and is thus identical to that used for the 10-pin RS232 interface of the DCM (see below.)

8.5 Interface box and DCM *NETWORK* ports (6-pin)

This is a standard 6-pin mil-spec plug (02E-10-06P).

Pin	Function
B	Data transmit +ve
C	Data receive +ve
E	Data receive –ve
F	Data transmit –ve

The cable from the *NETWORK* port to a standard RJ45 network connector is made up according to the following diagram.



8.6 Interface box *NETWORK* port (10-pin)

This is a standard 10-pin mil-spec plug (02E-12-10P).

Pin	Function
A	Power 0 V
B	Power +10 to +35 V
C	RS232 CTS
D	RS232 RTS
E	RS232 DTR
F	RS232 DSR
G	RS232 ground
H	RS232 CD
J	RS232 receive
K	RS232 transmit

The cable between this port and the attached DCM, *PORT A*, is a pin-to-pin 10-core cable:

8.7 DCM *PORT A* connector

The DCM connector is a standard 10-pin mil-spec socket (02E-12-10S).

Pin	Function

A	Power 0 V
B	Power +10 to +35 V
C	RS232 RTS
D	RS232 CTS
E	RS232 DTR
F	RS232 DSR
G	RS232 ground
H	RS232 CD
J	RS232 transmit
K	RS232 receive

8.8 *DM24 CONSOLE* and *AUTH CONSOLE* ports

This is a standard 6-pin mil-spec plug (02E-10-06P).

Pin	Function
A	RS232 transmit
B	RS232 receive
F	RS232 ground

8.9 Interface box *TAMPER* port

This is a standard 6-pin mil-spec socket (02E-10-06S).

Pin	Function
A	Tamper switch 0
B	Tamper switch 1
C	Tamper switch 2
D	Tamper switch 3
E	Tamper switch 4
F	Ground

Tamper switches should be connected between one of pins A – E and the common ground line F.

9 Sensor type codes

Sensor	Sensor type code	Units (V/A)
CMG-5T or 5TD, DC – 100 Hz response	CMG-5_100HZ	A
CMG-40T-1 or 6T-1, 1 s – 100 Hz response	CMG-40_1HZ_50HZ	V
	CMG-40_1S_100HZ	V
CMG-40T-1 or 6T-1, 2 s – 100 Hz response	CMG-40_2S_100HZ	V
CMG-40T-1 or 6T-1, 10 s – 100 Hz response	CMG-40_10S_100HZ	V
CMG-40, 20 s – 50 Hz response	CMG-40_20S_50HZ	V
CMG-40, 30 s – 50 Hz response	CMG-40_30S_50HZ	V
CMG-3T or 3ESP, 30 s – 50 Hz response	CMG-3_30S_50HZ	V
CMG-40, 60 s – 50 Hz response	CMG-40_60S_50HZ	V
CMG-3T or 3ESP, 60 s – 50 Hz response	CMG-3_60S_50HZ	V
CMG-3T or 3ESP, 100 s – 50 Hz response	CMG-3_100S_50HZ	V
CMG-3T or 3ESP, 120 s – 50 Hz response	CMG-3_120S_50HZ	V
CMG-3T, 360 s – 50 Hz response	CMG-3_360S_50HZ	V
CMG-3TB or 3V / 3ESP borehole, 30 s – 50 Hz response	CMG-3B_30S_50HZ	V
CMG-3TB or 3V / 3ESP borehole, 100 s – 50 Hz response	CMG-3B_100S_50HZ	V
CMG-3TB or 3V / 3ESP borehole, 120 s – 50 Hz response	CMG-3B_120S_50HZ	V

10 Revision history

2006-03-07	C	Added wiring diagrams; minor corrections.
2006-03-03	B	Revised and expanded
2005-09-05	A	New document