



CMG-DCM

Data Communications Module

Operator's guide

Part No. MAN-DCM-0001

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

Proprietary Notice: The information in this manual is proprietary to Güralp Systems Limited and may not be copied or distributed outside the approved recipient's organisation without the approval of Güralp Systems Limited. Güralp Systems Limited shall not be liable for technical or editorial errors or omissions made herein, nor for incidental or consequential damages resulting from the furnishing, performance, or usage of this material.

Issue B 2005-12-22

Table of Contents

1 Introduction.....	5
1.1 Inside the DCM.....	7
Serial port services.....	8
Network services.....	8
2 Installing the CMG-DCM.....	10
2.1 Overview.....	10
2.2 Power supply.....	10
2.3 Connecting to a single computer.....	11
Connecting over a serial link.....	11
Connecting over Ethernet.....	12
Using an internal modem.....	13
Connecting over USB.....	13
2.4 Connecting to a local network.....	14
Wi-Fi.....	16
Connecting to the Internet.....	16
2.5 Connecting digitizers and external hardware.....	17
2.6 Setting up the CMG-DCM.....	17
Web setup.....	18
3 Usage.....	21
3.1 General notes.....	21
3.2 The DCM as a data store.....	21
3.3 The DCM as a GCF data source.....	25
Accessing the DCM command line through gcf_out.....	26
Sensor arrays.....	27
3.4 The DCM as a network data hub.....	28
Communicating with digitizers.....	29
Data storage and retrieval.....	31
Real-time data transmission.....	33
3.5 Troubleshooting DCM installations.....	35
I cannot connect to the DCM's DATA OUT port using a terminal program.....	35
I cannot see the DCM's Web site over HTTP or HTTPS.....	36
I cannot connect to the DCM's Scream! server.....	36
I cannot GCFPING the DCM's Scream! server.....	36
I can GCFPING the DCM's Scream! server, but no data appears.....	37
The DCM is not receiving any data streams.....	37
The DCM is receiving streams, but they do not contain any data.....	38
The DCM receives streams, but gaps appear in the data some minutes after boot-up...	39
The DCM receives streams, but 2-minute gaps appear in the data at 4-hour intervals..	39

4 Configuration options	40
4.1 General	40
4.2 Disk	41
4.3 Serial port configuration	43
4.4 Network configuration	46
Ethernet port	46
Static routes	47
DNS setup	48
Incoming mail setup	49
Outgoing mail setup	50
Remote access	51
Administrators	52
PPP	53
mgetty	54
4.5 Data transfer	55
Scream!	55
Scream! Client	56
AutoDRM	56
HTTP server	57
HTTP client	57
CD1.0	57
CD1.1	58
DSS	60
SEED	62
5 Actions	65
5.1 Data Viewer	65
The icon bar	66
The stream list	66
5.2 Digitizer Setup	67
General digitizer settings	68
Digitizer output control	69
Trigger criteria	71
Auxiliary (“Mux”) channels	74
Sensor mass control	75
5.3 Disk tools	76
Partition, and format disk	76
Check disk filesystem	76
Flush flash	76
5.4 Disk files	77
5.5 Camera	77
5.6 Recent Log Entries	77
5.7 Summary	78
Network configuration	78
DNS configuration	79
Data Out Port, Serial Port A, Serial Port B	79
Flash Status	80

Tamper lines.....	80
Disk Status	80
Software Versions.....	81
6 Inside the DCM.....	82
6.1 File systems.....	83
6.2 Command line tools.....	85
6.3 Configuration.....	86
Digitizer console access.....	87
6.4 Monitoring.....	88
Data flow.....	88
Digitizer status.....	89
Tamper lines.....	90
6.5 Updating the DCM.....	91
Over the Internet.....	91
From the hard disk.....	91
Removing support packages.....	91
The firmware.....	92
Appendix A Connector pinouts.....	94
Appendix A.1 Modular DCM units.....	94
PORT A and B.....	94
DATA OUT port.....	94
USB connector.....	95
NETWORK connector.....	95
Appendix A.2 Integrated DCM units.....	96
DM/AM module output.....	96
Appendix B Sensor and digitizer types.....	97
Appendix B.1 Sensor response codes.....	97
Appendix B.2 Digitizer type codes.....	98

1 Introduction

The CMG-DCM is a versatile Linux-based module for storing and transmitting data captured using Güralp Systems Limited's range of seismic measuring equipment. Data can be gathered from up to three digitizers compatible with the Güralp Compressed Format (such as the Güralp CMG-DM24), and stored in its on-board Flash memory, from where it is written from time to time to a USB hard disk or to another device on your network. Once data is on the hard disk, you can connect to the DCM by various means to retrieve it.

Depending on your site requirements, the DCM may be supplied in several formats. Although these look different, they all share the same internal features.

- A *stand-alone* DCM, housed in a high-impact copolymer polypropylene case, may be connected to a digitizer through mil-spec connector cables and installed in a ground station or other location remote from the digitizer and seismometer. The stand-alone DCM can accept data from up to three devices connected to it through RS232 ports.



Stand-alone DCM units are normally supplied with high-capacity removable USB/FireWire disks for data storage. These disks allow the DCM to be used as part of completely autonomous installation in cases where communications links are poor or non-existent.



Retrieving data from a DCM's disk can be done by plugging it into a computer compatible with the USB or FireWire mass storage standards. Disks can be swapped at any time with no risk to data.

- A stand-alone DCM can also be supplied in a form factor suitable for mounting in a standard 19" rack. Such a DCM will normally come equipped with eight additional serial ports TTS0-7, and an optional State of Health (SOH) interface for monitoring or tamper evidencing.



Up to four modules may be mounted side-by-side in a 19" rack.

- An *integrated* or *modular* DCM may be combined with a digitizer and seismometer to produce a single unit capable of measuring seismic data, storing, and passing it on over a network:



This form of DCM uses an integrated 26-pin connector to communicate, rather than having separate ports.

A DCM in any of these forms may be supplied in the configuration most suitable to your requirements, depending on how you wish it to be connected to your network, and how its USB interface is to be used. The two sets of options are independent.

- Using the *10BaseT Ethernet* network option, you can connect the DCM to any TCP/IP network.
- Alternatively, an internal *PCMCIA modem* may be supplied, through which you can use a dial-up or other communications link to connect to your home network. Most 56k analogue and ISDN-type modems are supported.
- As a third option, an *802.11b Wi-Fi* module may be installed.
- If the *USB client* option is installed, you can connect the DCM's USB port to a computer and communicate with it as if it were on a private network.
- If the *USB host* option is installed, you can connect the USB port to an additional peripheral, such as a camera or external hard disk.

Whichever set of options you choose, you can use any free RS232 port to connect the DCM to a single computer for local monitoring and configuration, or through it to a wider network using PPP/SLIP.

1.1 Inside the DCM

The DCM's configuration is maintained by an internal database. All of the flexibility of the unit is provided through this configuration database. Before you deploy the DCM, you will need to configure it for your needs, either using its on-board Web server or over a direct serial link.

The two main areas which need to be configured determine the different services which the DCM provides from its serial and network ports.

Serial port services

You can use either the DCM's console or its Web configuration interface to configure the serial data ports. The *serial.x.service* configuration option determines which service each port provides.

- The simplest service is *getty*. This creates a console port, which you can use to log in to the Linux operating system of the DCM directly.
- The *mgetty-r* service is similar, but is more suited for serial links including modems, in which the link is not always active. This is the default option.
- Any of the serial ports on the DCM can be directed to transmit GCF (Güralp Compressed Format) data by setting the *serial.x.service* option for that port to *gcf_out*. You can then run *Scream!* or other software on your computer to receive the data. You will not be able to access the Web configuration interface using this service, although you may be able to configure attached digitizers from *Scream!*.
- Likewise, any serial port can be set to listen for incoming data (*e.g.* from a connected digitizer or another DCM) by setting the *serial.x.service* option to *gcf_in*. The DCM will automatically combine this data with any other streams it is receiving, and send it on using all *gcf_out* ports and any other transmission methods which have been configured.
- You can log in to the Linux operating system over a modem link by setting the *serial.x.service* option to *mgetty*. *mgetty* listens for incoming calls from your computer and sets up a login session for you. You may need to configure *minicom* or *hyperterm* to communicate with your own modem before you can do this.
- Both *mgetty* and *mgetty-r* can recognize incoming PPP connections and will automatically run a PPP daemon if you attempt to connect to the DCM in this way.

Network services

Because the DCM is a fully-fledged Linux machine, you will need to set up networking before you can use it. Advanced networking is beyond the scope of this manual. For a basic setup, you will need to perform the following actions:

1. Assign the DCM an IP address. By default, the DCM will try and use DHCP (Dynamic Host Configuration Protocol) to find its own IP address. However, this requires a DHCP server on your network. If you

do not have a DHCP server, you will have to set the IP address yourself.

2. Find out the IP address the DCM is using (if you have not set it yourself).
3. If necessary, configure your own computer's IP address so that it can communicate with the DCM.

See [Section 2.4, "Connecting to a local network"](#) for full details.

Once you have connected the DCM to your local network or to the Internet, you can use a wide range of methods to retrieve data from it.

- All DCM units feature an on-board Web server, which allows you to monitor and configure the station from any browser (and download data by HTTP, if enabled.)
- For maintenance, you can log in to the operating system directly over the secure shell protocol, SSH, and use all the standard Unix/Linux tools.
- A GCF server can be set up to transmit data to any application that supports the GCF format (such as Scream!, Antelope, or Earthworm.)
- The DCM can act as a CD1.0 or CD1.1 transmitter to a specified client, or to multiple clients as an option.
- Optionally, the DCM can also receive and process AutoDRM messages.

In addition, the DCM can be configured to act as a data client, receiving GCF and CD1.0/1.1 data streams, combining these streams and storing or retransmitting them as appropriate. See [Chapter 3, "Usage"](#) for full details.

2 Installing the CMG-DCM

2.1 Overview

The DCM is effectively a low-power Linux computer dedicated to seismic data flow tasks. Resources permitting, the DCM can perform any computational task you require. Because of this flexibility, the DCM must be configured for your particular purposes before it can be used:

1. If you only need to use the DCM as a data store, you can gain console access over the serial port for configuration.
2. If you are going to use the DCM as part of a TCP/IP network, you must set up its IP address and other networking parameters before you can connect to it. Whilst the network is inoperative, you will need to connect to the DCM console over a direct serial connection. We recommend that you configure the DCM correctly, to the best of your knowledge, *before* shipping it into the field.
3. If you want to connect to the DCM using TCP/IP over modem links, you must set up PPP over these links. Configuring PPP is beyond the scope of this manual: please see the Linux man page for `ppp` for details, or the Linux PPP HOWTO.
4. Once communication with the DCM is working, you should set up the serial ports on the DCM to provide the services you require.
5. When all ports are configured, you can use the DCM to set up attached Güralp digitizers and digital instruments.

2.2 Power supply

The *stand-alone* DCM receives its power from lines in its data connectors. All three of these ports are Güralp standard 10-pin combined serial/power interfaces. Cables are available from Güralp Systems which enable you to connect a 9-pin D serial interface and 2-way power connector to any of these ports.

You can power the DCM from whichever data port is most convenient, whether or not it is being used for receiving or transmitting data. However, you should always supply power to the *DATA OUT* plug if possible, since it is easy to accidentally short the pins of a male connector. Attached digitizers and sensors are powered from the same supply as the DCM. Any 12 V DC power supply may be used; if using batteries, we recommend that you use a unit with low-voltage disconnect to avoid excessive drain.

A *rack-mounted* DCM has a separate *POWER IN* connector on the back panel, which should be attached to a 12 V DC supply. There is also a *POWER OUT* connector on this panel, which allows you to daisy-chain several DCM units together. The *POWER OUT* connector is provided merely for convenience: lines from the *POWER IN* connector pass straight through.

An *integrated* DCM receives its 12 V DC power from two pins in its single connector. See [Appendix A, “Connector pinouts”](#) for the positions of these pins.

2.3 Connecting to a single computer

In autonomous or temporary installations, you will only need to connect the DCM to one computer (*e.g.* a laptop) for initial configuration. Follow these instructions to make the DCM communicate with your computer.

Note that your computer must either have an RS232 (serial) port for initial connection, or be running its own DHCP server. DHCP servers are available for Windows, but are not supplied as standard.

If you want to connect the DCM to a local area network, follow the instructions in [Section 2.4, “Connecting to a local network”](#) instead.

Connecting over a serial link

The *DATA OUT* port can be used to connect a DCM to a single computer through a direct serial link. This link can be used to log in to the Linux operating system of the DCM and transfer files. It is most useful for maintenance and troubleshooting over a low-bandwidth connection, and for initial setup.

To communicate over a serial link:

1. Connect an RS232 reverse serial cable between the DCM and your computer. A suitable cable should have been supplied with the unit.
2. Run a terminal emulation program on the computer. The built-in programs `minicom` for Linux, and `hyperterm` for Microsoft Windows, can be used for this.
3. Configure the baud rate of the serial link. By default, the DCM uses a baud rate of 115200, with 8 data bits, no parity bit, and one stop bit, and *without* flow control.
4. Log in with your username and password, or `root` if you have not yet created one. See <http://www.fags.org/docs/Linux-HOWTO/Text-Terminal-HOWTO.html> for full information on how to set up terminal emulation under Linux.

5. You now have access to the DCM's console and all of its functions. You should change your password, if you have not done so already, with the command `passwd`

For full details on how you can configure the serial ports of the DCM, see [Section 4.3, “Serial port configuration”](#).

Connecting over Ethernet

You can use a “crossover” Ethernet cable to connect the DCM directly to your computer. This has the same effect as connecting the DCM and your computer to a separate hub using standard Ethernet cables: the two machines will constitute a network segment, and each will have its own IP address on that segment. See [Section 2.4, “Connecting to a local network”](#) for details on how to configure TCP/IP on the DCM.

If your computer has two network interfaces, and one is connected to a local area network, you may be able to set up a network bridge between the DCM's segment and the rest of the network. For example, if you are using Microsoft Windows XP:

1. Connect to the DCM's console over a serial link.
2. Because you are connecting the DCM to a single computer, you will probably not be able to use DHCP. You should configure the DCM to disable DHCP and use an IP address in a private range such as `192.168.0.x`:

```
gcfgdbset net.eth.0 static
gcfgdbset net.eth.0.address 192.168.0.2
gcfgdbset net.eth.0.netmask 255.255.255.0
```

Here, `192.168.0.2` should be replaced with the IP address you want the DCM to use.

Allow a short time for the DCM to reconfigure the network.

3. On your computer, select **Start** → **Control Panel** → **Network Connections**. Right-click on the interface connected to the DCM, and select **Properties**.
4. Click on **Internet Protocol (TCP/IP)**, then **Properties**.
5. Select **Use the following IP address**, and fill in an address on the same subnet as the DCM (e.g. `192.168.0.49`). Click **OK**, then **OK** in the *Connection Properties* window.
6. Check that you can connect to the DCM using its new IP address by

opening its Web site at

`http://192.168.0.2/`

7. *Advanced usage:* If you later want to connect your computer to a local area network, you can make the DCM visible to that network by setting up a network bridge. *Before* doing this, you must make sure that the IP addresses you have chosen for the DCM and your computer are suitable for the local network you want to connect it to.

To set up the bridge, open the *Network Connections* window, and select the connections corresponding to the DCM and your network. Click **Bridge connections**, and follow the instructions in the wizard.

If you are using Linux or another operating system, you should see its own documentation for more details.

Using an internal modem

If you have ordered a DCM with an internal PCMCIA modem, it will automatically listen for incoming PPP connections. This is most useful if the DCM is integrated into a borehole instrument, which can then be connected to a modem at a central data collection point. The modem within the DCM appears on the Web configuration interface as a serial port, allowing you to change the service it provides and the data transfer settings it uses. For example, if you do not want the internal modem to use PPP, you can configure it to use *mgetty* instead and log in directly.

Connecting over USB

If your DCM is equipped with the USB *client* interface, you can connect it directly to any computer with a USB socket. When this is done, the computer treats the module as if it were a standard network card. For example, to connect the DCM to a PC running Microsoft Windows, you should:

1. Link the DCM's USB port to any available USB socket on the computer.
2. After a short wait, Windows should report that a new USB device has been detected. When prompted to install drivers for it, you should use the ones provided on CD by Güralp Systems.

Once the “network card” is installed, the two machines will constitute a network segment, and each will have its own IP address on that segment. See [Section 2.4, “Connecting to a local network”](#) for details on how to configure TCP/IP on the DCM. If you wish, you can set up a network bridge between this connection and a local area network or the Internet as described above.

2.4 Connecting to a local network

The DCM is normally supplied with a 10BaseT Ethernet port. This port can be used to connect to your local network.

When the DCM starts up, it will try to find a DHCP server on your network to assign it an IP address.

If you use DHCP on your network, and you want to access the DCM's Web server or console over the network, you may be able to find out from the DHCP server which address it has given the DCM, in which case you can access it directly. Otherwise, you will need to connect to the DCM over a serial link to find out its IP address.

If you do not use DHCP on your network, you will have to set the DCM to use a static IP address.

1. Make a temporary serial connection to the DCM from a local PC. You should see the message

```
DCM login:
```

2. Enter `root` and the administrator password. If you have not been given a password, the default setting is `rootme`. The DCM will reply with a prompt:

```
[root@DCM ~]#
```

This verifies that the unit is working properly. You should change the password as soon as you can with the command `passwd`

3. Connect the DCM's *NETWORK* port to your network, and power cycle it. Log in again.
4. If your network *does not* use DHCP, type

```
gcfgdbset net.eth.0 static
```

to make the DCM use a static IP address instead, and set the address with

```
gcfgdbset net.eth.0.netmask 255.255.255.0  
gcfgdbset net.eth.0.address your-address
```

Here, your-address should be replaced with the IP address you want the DCM to use. The IP address you choose must be unique on your network.

If you are connecting the DCM to a machine on the same network, you do not need to configure any more options at this stage.

If your network uses DHCP, you will not need to perform this step. Instead, ensure the DCM is also using DHCP with

```
gcfgdbset net.eth.0 dhcp/bootp
```

5. Issue the command `ifconfig`

The DCM will reply with technical information on its current network setup. Each interface is listed separately. The *NETWORK* port uses the interface `eth0`:

```
eth0      Link encap:Ethernet  HWaddr 08:00:20:C0:FF:E2
          inet addr:192.168.48.187  Bcast:192.168.48.255
Mask:255.255.255.0
          UP BROADCAST NOTRAILERS RUNNING MULTICAST
MTU:1500  Metric:1
          RX packets:88045 errors:36 dropped:0 overruns:36
frame:0
          TX packets:16308 errors:0 dropped:0 overruns:0
carrier:0
          collisions:149 txqueuelen:100
          RX bytes:12358214 (11.7 MiB)  TX bytes:0 (0.0 B)
          Interrupt:76
```

If the interface is correctly set up, its IP address will be shown after `inet addr:` (here, 192.168.48.187.)

6. If you are connecting the DCM to a single computer which does not run a DHCP server, you will need to configure that computer to use an IP address on the same subnet as the DCM before they will be able to reach each other (see [“Connecting over Ethernet” in Section 2.3.](#))

You should now be able to connect to the DCM's Web setup interface by typing its IP address into any browser, *e.g.*

<https://192.168.0.2/>

Scream! and similar software applications should also be configured to use this address to communicate with the module, if they are intended to communicate over the network.

Alternatively, you can continue using the `gcfgdbset` command to set other configuration options by name. In [Chapter 4, “Configuration options”](#), each option's name is given in *italics*. The command syntax to use is

```
gcfgdbset option-name new-value
```

The options will take effect immediately whenever possible. Some changes may take some time to complete, since services may need to be restarted. The `gcf gdbset` command performs only simple checks on the new value, so you should check the syntax of the option carefully.

Wi-Fi

Optionally, a DCM may be supplied with 802.11 (“Wi-Fi”) hardware in place of Ethernet. There are two modes in which an 802.11 network may operate:

- In *ad hoc* mode, data packets are sent out over the wireless connection indiscriminately and are received by all machines on the network. This is the simplest style of Wi-Fi network, but consumes more power and cannot easily be extended to large networks.
- In *infrastructure* mode, dedicated 802.11 hubs collect data packets, collate, and distribute them efficiently around the network. These hubs may also act as transparent extensions to existing (wired) TCP/IP networks.

The DCM is capable of running in either situation, depending on your requirements. As far as the computers on the network are concerned, there is no difference between a TCP/IP network running over Ethernet and one using 802.11 connections, or a mixture of the two.

Once the 802.11 system is running, you will need to run a DHCP service on your network, or manually set the IP address of the DCM as above.

Connecting to the Internet

If the local network is already connected to the Internet through some other means (perhaps through a gateway machine), you can create a default route with

```
gcf gdbset net.eth.0.gateway network-gateway
```

where network-gateway should be replaced with the IP address of the gateway on your local network.

The DCM can perform more complex routing if required. See [“Static routes” in Section 4.4](#) for more details.

As supplied, the DCM module will accept requests to its on-board Web server and logins over SSH. In addition, the DCM can be requested by Scream! or other GCF-compatible software to send GCF streams to your computer.

If the instrument is located on a private network, you may be able to connect to it from the wider Internet by using a feature implemented by SSH known as *tunnelling*. You can use this technique to connect to the DCM through a chain

of intermediate machines which support SSH. Once this chain is set up, you can treat the connection as if it were a direct link between the DCM and your computer. Many standard Internet protocols may be fed through SSH in this manner. Whether this is possible will depend on the precise configuration of your local area network. For more details, please see the documentation for SSH clients such as `ssh` and `putty`.

2.5 Connecting digitizers and external hardware

The DCM is designed for use with Güralp Systems digitizers, which communicate over RS232 or RS422 serial links using the GCF protocol. There are three RS232 ports available on a stand-alone DCM and ten on a rack-mounted model, whilst on a DCM bonded to a digitizer unit only the *DATA OUT* port is available (the digitizer is connected directly to the *PORT A* interface).

The three serial ports on a stand-alone DCM are labelled *DATA OUT*, *PORT A*, and *PORT B*. In a typical setup, one or both of the latter two ports are connected to Güralp DM-24 digitizers, whilst the *DATA OUT* port connects the DCM to a computer or serial modem. However, the labels are provided merely for convenience: internally all three serial ports behave identically. Thus, if the DCM is connected to a network over Ethernet, you could use all three ports to communicate with digitizers. Conversely, if you have only one digitizer attached to a stand-alone DCM, either or both of the remaining two ports could be used to transmit data gathered by the module.

A rack-mounted DCM has a *CONSOLE* port with a female connector, corresponding to the *Data out* port of the stand-alone model, and nine serial ports (with male connectors) for communication with digitizers. These appear on the Web configuration interface with their Linux device names `/dev/ttyS0`, `S1`, etc., up to `S7`. The *TTY* port corresponds to `/dev/ttySB0`, whilst the *CONSOLE* port is shown as `/dev/ttySA0`.

If the DCM has a USB *host* interface, you can connect it to any USB peripheral supported by its Linux operating system. For example, you may want to attach a camera to the DCM, or an additional USB mass storage device. If there is no hard disk inside the module, it will automatically search for suitable storage on the USB interface. The DCM may alternatively have been supplied as a USB *client*; if this is the case, the USB connection can be used to connect the module to a single computer (see below).

2.6 Setting up the CMG-DCM

Once a DCM is installed, it must be configured to your particular requirements. The principal way to set up a DCM is through its on-board Web server, which

gives you access to all the module's configuration options. You can use this Web interface over any network connection using the HTTP or HTTPS protocols. Alternatively, you can log in to the DCM's Linux operating system and issue commands directly: see [Section 6.2, “Command line tools”](#).

Web setup

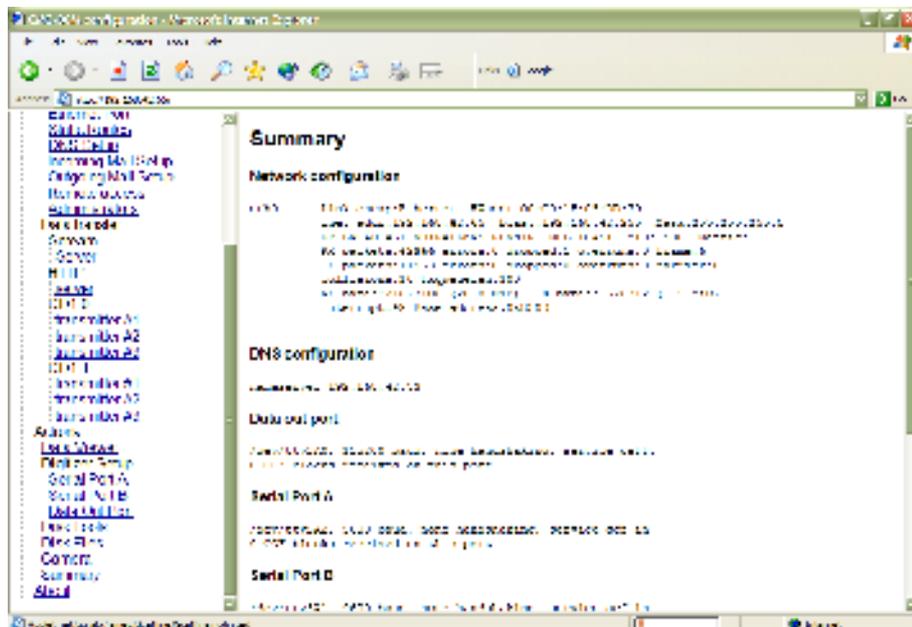
The DCM provides a Web interface which you can use to set up the system. To access this from anywhere on the local network, open any web browser and navigate to the IP address of the DCM. For example, if the DCM has the address 10.82.0.1 you would enter

<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 DCM as supplied accepts connections only over *https*, although this may be altered if required (see [“HTTP server” in Section 4.5](#))

Your browser may ask you for a username and password to access the Web configuration interface. An initial *root* username and password will have been supplied by Güralp Systems; however, you should check with the DCM's administrator what username you should use. On a clean install of the Linux distribution, the password for *root* is *rootme*.

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 DCM'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.

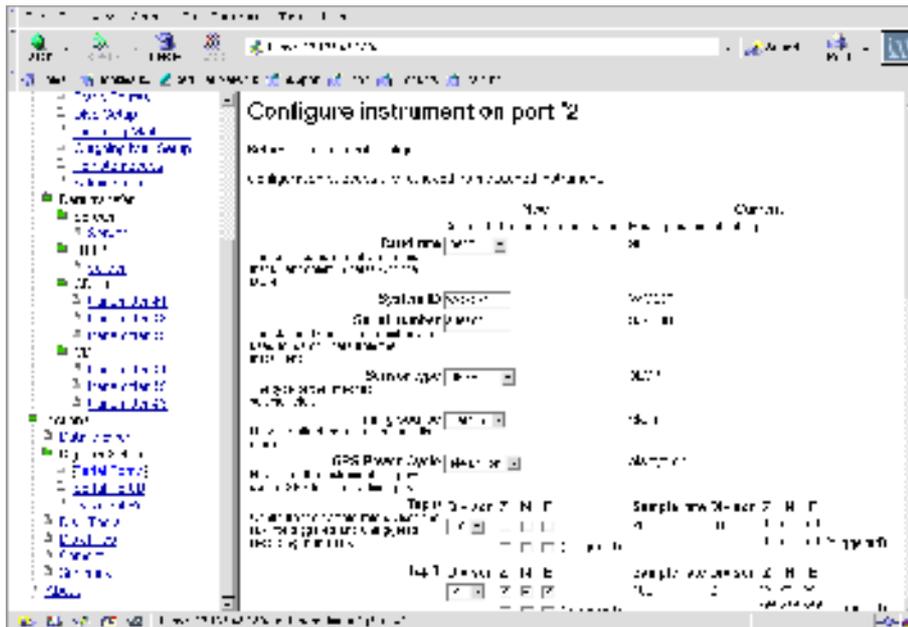
The links in the menu on the left lead to various pages where you can change the configuration of the DCM. Initially, you should be sure to change your administrator password. Click on *Administrators* under *Network configuration*, fill in the new password in the two text boxes below *Existing accounts*, near the bottom of the page, and click *Change password*. If you want to add any other named accounts, you may do so at this stage (see [“Administrators” in Section 4.4](#)).

At this point, you should configure the DCM for your own needs. Examples of how to configure a DCM for several common applications are given in [Chapter 3, “Usage”](#), whilst a full reference is provided in [Chapter 4, “Configuration options”](#). Some important points to note are:

- The DCM 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 DCM unreachable over the network. You should review carefully any changes you make to these settings before committing them.
- You can disable the DCM's *ssh*, *http*, and *https* servers using the links under *Data transfer*. Before doing this, you should ensure that you will still have access to the DCM by other means, *e.g.* by logging in directly over *ssh*.

Setting up digitizers

You can change the configuration of attached DM24 digitizers through the DCM's web page interface. To do this, click on *Digitizer Setup* beneath *Actions* in the menu tree. The DCM will then retrieve the current configuration from the DM24, which will take a few seconds. This done, a page will appear allowing you to alter the digitizer's settings:



A useful feature of Güralp Systems digitizers is the ability to output streams at several data rates. Each stage of filtering is known as a *tap*, producing data at a particular rate. At each tap, incoming data is filtered and its data rate reduced by a user-selectable factor of 2, 4, 5, 8, or 10 (in the case of Güralp DM24 digitizers). The data rate at each tap must be an integer number of Hz. For a full listing of the possible tap configurations, see [Section 5.2, “Digitizer Setup”](#).

The initial digitizing process outputs data at a rate of 2000 Hz, which is decimated at Tap 0 by a fixed factor of 10. You can control which data streams the digitizer outputs at this point using the top row of the first grid of *Z*, *N*, and *E* checkboxes on this page. Subsequent grids of checkboxes determine which of the other taps output data. For example, you could output the *Z* component at 200 Hz, together with all three components at a lower rate.

You can also set trigger criteria for the digitizer. The criteria can be applied to any component at any tap, and as soon as they are satisfied, any component with a checked box in the lower (*triggered*) row of one of the grids will begin outputting data in addition to the normal streams. You cannot have the same component produce both triggered and continuous output at the same tap.

See [“Digitizer output control” in Section 5.2](#) for full details of what you can do.

Using Scream!

If you prefer, you can use Güralp Systems' Scream! software to configure and calibrate digitizers and sensors through the DCM module, just as if they were connected directly. For full details on how to use Scream!, please refer to its own documentation or the extensive on-line help.

3 Usage

The DCM can be integrated with any system where seismic data needs to be collected, collated, or converted from one form into another. It is designed to operate as transparently as possible, and once connected and configured for a particular role in a system it should not require further maintenance.

The rest of this chapter gives detailed installation and usage notes for several common DCM installations. Between them, they highlight many important features of the DCM. For full configuration information, please refer to [Chapter 4, “Configuration options”](#).

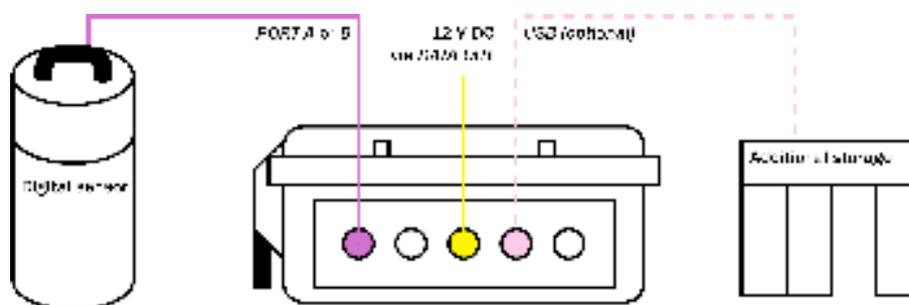
3.1 General notes

Stand-alone DCM modules can be supplied with a touch-screen and minibrowser as an option. This allows you to perform configuration tasks on-site. However, the touch-screen imposes some environmental restrictions on the unit.

For all other DCM units, you will need console or network access to the DCM to configure it for your installation. This is especially important if the DCM is not to be part of a TCP/IP network whilst in use. As supplied, the *DATA OUT* port runs the *getty* service, which you can use to access the console of the DCM over a serial link. Alternatively, you can temporarily connect a computer to the *NETWORK* port. It is recommended that you keep a connection to the DCM available for maintenance and troubleshooting, even if the link is too slow for transferring data.

3.2 The DCM as a data store

The simplest way to use a DCM is as a storage medium for digital data.



1. Connect the DCM to a digitizer or digital sensor.
2. Connect the *DATA OUT* port to a source of 12 V DC power.

3. Connect the *NETWORK* socket to a computer as described in [Section 2.4, “Connecting to a local network”](#) and perform any necessary configuration of the DCM and digitizer. You will be prompted for a username and password: log in as `root` and use the password supplied by Güralp Systems.
4. The DCM will immediately begin recording data into Flash memory as it is received, and every so often data will be moved onto the internal hard disk. At this point you can leave the DCM running without assistance.

Data is stored in the Flash memory as a number of files in Güralp Compressed Format (GCF) unless you have specified another option. Once the memory becomes 75% full, files are automatically moved to the hard disk until it is less than 50% full. (See [Section 6.1, “File systems”](#) for details of the algorithm used.)

You can check that the DCM is receiving data either by monitoring the *Summary* page of the on-board Web interface (see [Section 5.7, “Summary”](#)), or from a command prompt using the command

```
gnblocks port-number
```

to find the number of GCF blocks received so far on a port. To see the number of blocks recorded on all ports, type

```
gnblocks
```

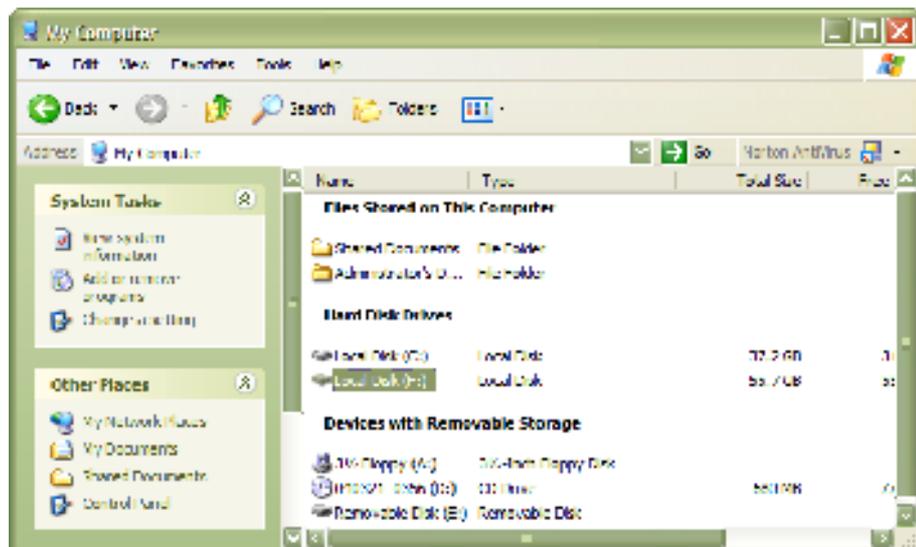
5. If the internal hard disk is missing or becomes full, and you have chosen the USB *host* option, the DCM will automatically look on the USB interface for an external USB mass storage device.
6. To replace the internal hard disk, unclip the cover of the DCM and hold down the lever button to bring the disk out from its housing.



Slide the disk out and replace with another Lacie U&I drive, or any brand of IDE / USB or IEEE 1394 2.5" drive you specify (at the time of manufacture). You can do this at any time without losing data.

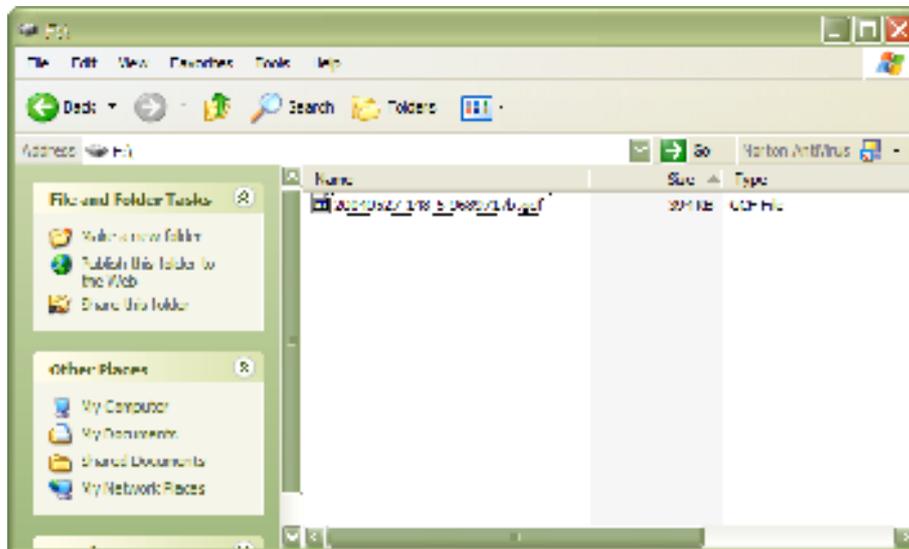
7. Plug the hard disk into any computer that supports the USB Mass Storage standard using a standard USB cable. Newer Linux distributions and Microsoft Windows XP have this enabled by default.

In Windows XP, you should see a series of **Found new hardware** messages indicating that the drive has been recognised. A new disk drive icon should appear in *My Computer*:



This process may take several minutes to complete.

8. Double click on the drive's entry to browse the files inside and copy them to your data store.

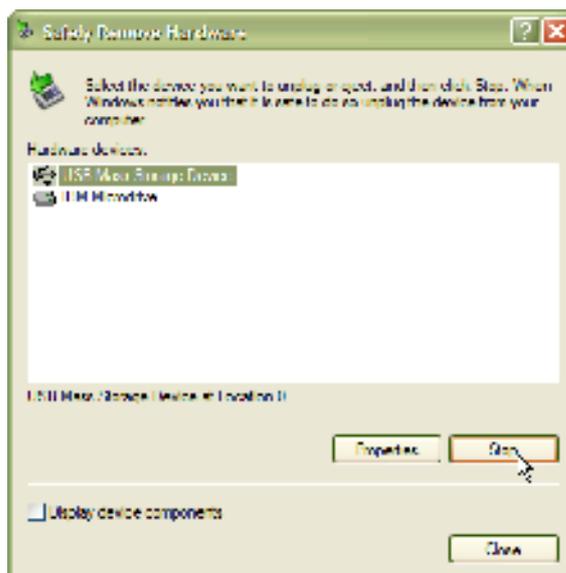


Alternatively, you can open the files directly from the USB disk using Scream! or other GCF-compatible software.

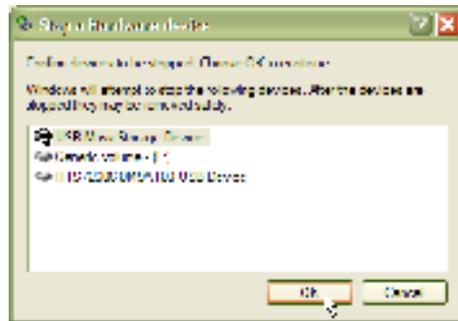
9. When you want to remove the USB disk, double-click on the **Safely Remove Hardware** button:



10. Choose the USB port attached to the disk, and click **Stop**.

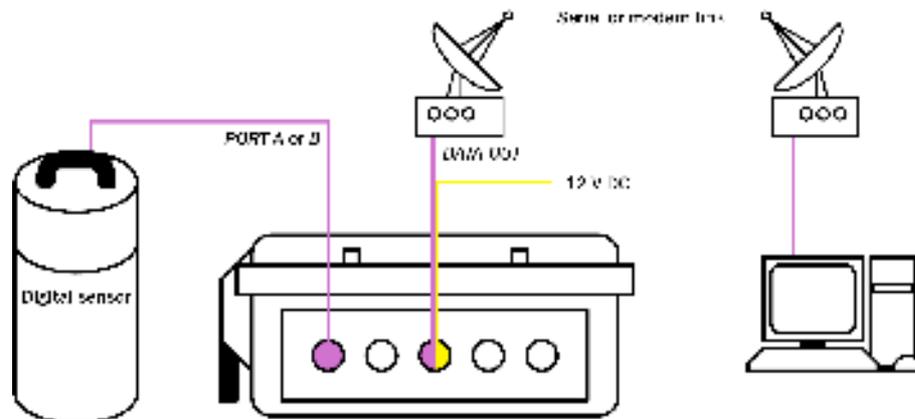


11. In the next window, check that the correct hardware is shown, and click **OK** to confirm.



12. You can now remove the hard disk from the computer and reinstall it in the DCM if required.

3.3 The DCM as a GCF data source



With a sufficiently fast serial link, you can instruct the DCM to send incoming data directly to a GCF-compatible client. For example, Güralp Systems' *Scream!* software allows you to display and record incoming data, as well as change the settings of attached digitizers. You can do this in addition to recording data on a local hard disk, or you can leave the hard disk uninstalled and operate the DCM entirely over the network link.

This example shows a module communicating with a local PC over a dedicated radio link. You could also use a simple serial cable to connect the DCM to the PC.

1. Connect the DCM to a digitizer or digital sensor.
2. Connect the *DATA OUT* port to a source of 12 V DC power.

3. Either use a computer connected to the *NETWORK* port to browse to the *Serial port configuration* section of the DCM Web site, or log in to the Linux console over the serial link.
4. Set the *serial.2.service* option to *gcf_out*. If using the Linux console, this can be done using the command `gcfgdbset` (see [Section 6.2](#), “[Command line tools](#)”.)

Note: Beyond this point you will *not* be able to use the serial link to access the Web configuration interface of the DCM. If you do want to be able to do this, you should configure the serial link for PPP (see “[PPP](#)” [in Section 4.4](#)) and run a Scream! *server* on the DCM. You will need to assign the DCM its own IP address on your local network.

5. If you connected to the DCM by PPP, you will lose the network connection at this point, because the DCM is now using the serial link directly.
6. Open Scream!'s main window, and look under *Local* in the tree on the left for the serial port which is communicating with the DCM. If it is not shown, you may have to 'unfold' the tree to reveal it. The digitizer(s) attached to the DCM will appear underneath the entry for the serial port, and you can configure them from within Scream!. See Scream!'s documentation for more details.
7. If no digitizer is shown, it may be that your software is not configured correctly for the serial link. As supplied, the DCM uses a baud rate of 115200 on its *CONSOLE* port, with 8 data bits, no parity bits and 1 stop bit, and no flow control.

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.

Accessing the DCM command line through `gcf_out`

The *gcf_out* service provides a command-line terminal designed to be compatible with Güralp Systems' SAM serial storage device. To access this terminal from Scream!, right-click on the DCM's icon and select *Terminal...* The SAM commands you can use are described in the SAM Operator's Manual.

You can also log in to the DCM's Linux operating system through the terminal. Enter the command

```
GETTY
```

and press ENTER. You will be presented with the `login:` prompt.

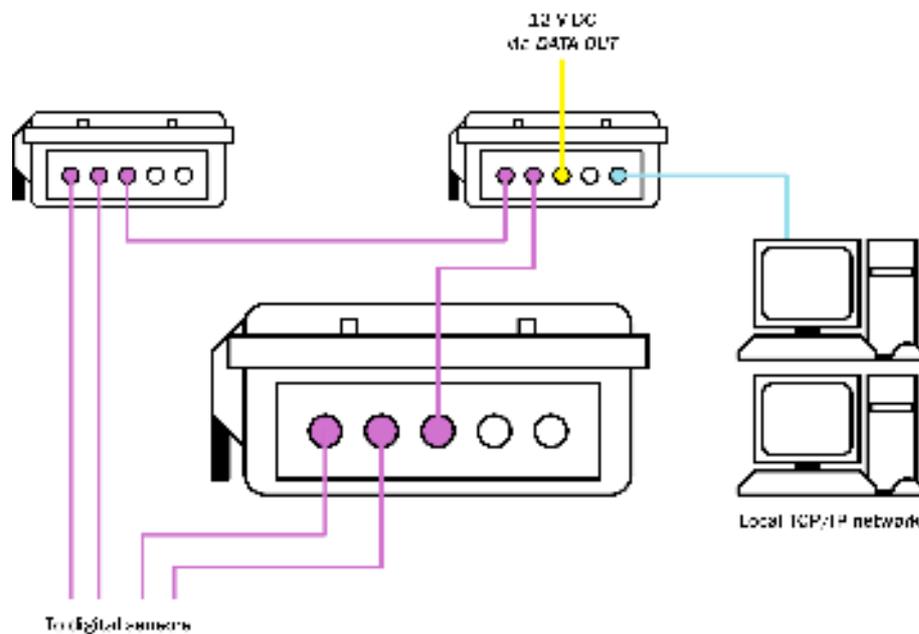
If you are not using Scream!, you can interrupt the GCF data stream and drop

straight to the `login:` prompt by connecting direct to the serial port and typing `forcegetty`

It may take several attempts for this to succeed.

Sensor arrays

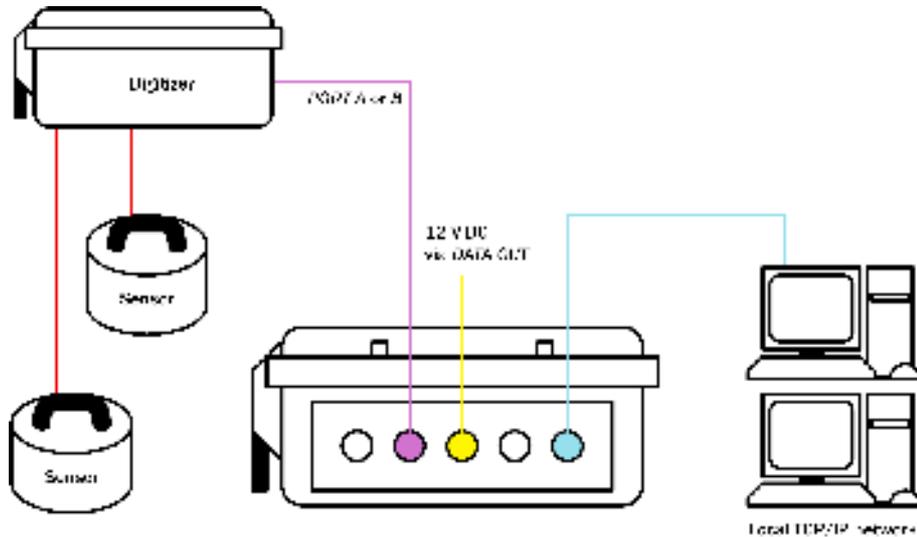
Another possible reason for setting the *DATA OUT* port's service to *gcf_out* is to allow several DCM units to work together and aggregate the inputs from an array of sensors.



Here, *DATA OUT* ports with *gcf_out* as the service output data to the serial inputs of another DCM or AM module, which aggregates them all and sends them out over a network connection.

The DCM can handle multiple data sources with ease using the *gcf_out* service. In practice, however, it is almost always preferable to set up a local area network and use PPP over any serial links, so that GCF, Web and SSH traffic can share the same connection. Setting up a fully networked array is slightly more involved than using the simple *gcf_in* and *gcf_out* services, but affords much more flexibility. See below for more details.

3.4 The DCM as a network data hub



The most flexible way to operate the DCM is as a fully independent machine on your local area network. To do this:

1. Connect Güralp digitizers to *PORT A* and *PORT B* as necessary, using the serial data cable provided.
2. If you are using the DCM's USB capabilities, connect your external hardware or computer to the *USB* socket.

Note: USB distinguishes between *hosts* and *clients*. The DCM can act as either, depending on the options you specified at manufacture. If your DCM is a *host*, you can attach external USB peripherals such as hard disks. If it is a *client*, you can attach it to a computer, and it will appear as a network interface.

3. Connect a Güralp combined serial/power cable to the *DATA OUT* port. Make up a connector if necessary, and attach the power lines to a 12 V DC power supply.
4. Connect the 9-pin serial socket to a computer for configuration.
5. Connect an Ethernet cable to the *NETWORK* socket, and set up the network as described in [Section 2.4, “Connecting to a local network”](#).

You should now be able to connect to the DCM's Web setup interface by typing its IP address into any browser, *e.g.*

<https://192.168.0.2/>

By default you can use either *http:* or *https:* URLs to access the DCM's Web site. HTTPS is a secure variant of HTTP, which we recommend you use in preference to avoid passwords being sent over the network in clear text. Scream! and similar software applications should also be configured to use this IP address to communicate with the module. If the DCM is using a static IP address, you may assign it a name on your network (although you will still need to use the numeric IP address in Scream!.)

The DCM also runs a SSH (secure shell) server, which you can use to access its Linux command line over the network, just as if you had connected to it over a direct serial link. The standard Linux program `ssh` and the freeware Windows program `putty` are popular SSH clients.

If you do not wish to use the DCM's Web site, you can continue using the `gcfgdbset` command to set other configuration options by name. For the option names, you should refer to Chapter 4, "Configuration options" in the DCM manual, where each name is given in *italics* with a description of its action.

The command syntax to use is

```
gcfgdbset option-name new-value
```

The options will take effect immediately whenever possible. Some changes may take some time to complete, since services may need to be restarted. The `gcfgdbset` command performs only simple checks on the new value, so you should check the syntax of the option carefully.

Communicating with digitizers

From the DCM, you can connect to the console of attached digitizers with the command

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

Alternatively, you can send individual commands to a digitizer with the command

```
gcli port-number command
```

In these commands, *port-number* is the port number and *port-name* the port device name to send the command to. To find out these, issue the command `serialmap`:

```
Library version: libserialmap Version 1.0.5 with LIBGCONFIGDB
3 serial ports
  Port 0, Key 7000, name Data out port, device /dev/ttySA0, baud
115200
  Port 1, Key 7001, name Port B, device /dev/ttySA1, baud 9600
```

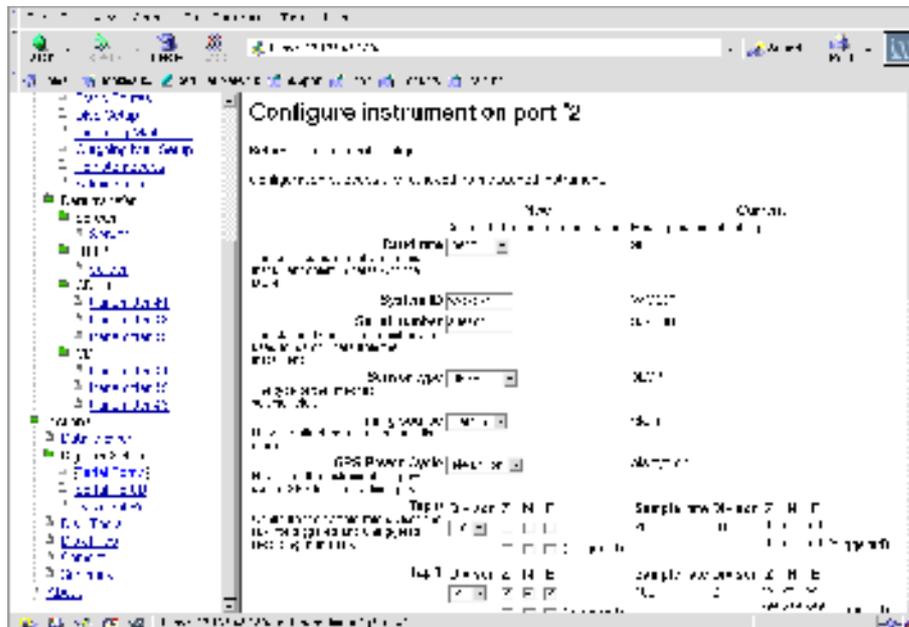
Port 2, Key 7002, name Port A, device /dev/ttySA2, baud 19200

Each port will be listed, with its number (e.g. 0), name (e.g. Data out port), and device (e.g. /dev/ttySA0).

Configuring digitizers

You can configure digitizers directly using the above commands. However, the DCM also provides a Web interface to the digitizer's configuration options.

1. Choose the entry for the relevant port under **Actions** → **Digitizer Setup** on the menu tree.



2. You can change all of the digitizer's configuration options from this page. If you have attached a 6-channel digitizer, the page will reflect this:

Click **Save changes** to save the configuration and start writing SEED volumes. One volume will be written for each stream every three hours. The DCM will always start a new volume if it is power cycled; the aborted file will still be a valid SEED volume.

Real-time data transmission

You can use the *DATA OUT* port (or any of the DCM's serial ports) in several ways, depending on the *service* you have chosen to run on the port.

- By default, the *DATA OUT* port runs the *getty* service, which allows you to connect to the DCM's console.
- The *mgetty* and *mgetty_raw* services are similar, but can deal with connections made through modems.
- The *gcf_out* service causes the DCM to combine the incoming GCF streams and send them on transparently, as for a Güralp SAM or CRM.
- The *gcf_in* service turns the port into an extra digital data input port. The DCM's *PORT A* and *PORT B* run the *gcf_in* service.

Scream! server

In a straightforward vault installation, we recommend that you leave the *DATA OUT* port running the *getty* service for emergency console access, and instruct the DCM to act as a server for incoming data on your network. This can be done from the **Data transfer** → **Scream!** → **Server** page on the Web site:

1. Enable the Scream! server on the DCM by setting *datatransfer.scream.server* to *on* and *datatransfer.scream.server.allowserialaccess* to *yes*. Make a note of the port number *datatransfer.scream.server.port*, or change it to another. Click **Save settings**.
2. Start up Scream!, and select **Windows** → **Network Control...** from the main menu. Click the **My Client** tab.
3. If it is not already selected, check **Receive Data** to start Scream! listening.
4. Right-click anywhere in the **Servers** list box, and select **Add UDP Server...** Enter the IP address and port number of the DCM, separated by a colon : (*e.g.* 192.168.0.2:12345)
5. Test communications by right-clicking on the newly-added server, and selecting **GCFPING**. A message appears in the *Control* pane logging the ping being sent. If communication is good, and the server is enabled

for client requests, you will receive a *GCFACKN* message from the server which will also appear in the *Control* pane.

6. Request data by right-clicking on the server and selecting **GCFSEND:L** (or **GCFSEND:B**) from the pop-up menu. (**L** is used for little-endian and **B** for big-endian byte order, and are distinguished for compatibility.) Streams should soon begin to appear in *Scream!*'s main window.
7. To stop the link, right-click as before and select **GCFSTOP** from the pop-up menu. If you do not **GCFSTOP**, the server will continue to transmit to a client that is no longer listening. You should ensure that the server replies with a *GCFACKN* message. If an acknowledgement does not appear in the *Control* pane, repeat the **GCFSTOP** command.

CD1.0 and CD1.1

The DCM can be configured to transmit incoming data to a specified client using the CD1.0 or CD1.1 protocols. If you intend to send CD1.0 or CD1.1 data to a National Data Centre (NDC), the installation will need to be assigned a unique code by the International Seismic Centre or with the US's National Earthquake Information Center.

To enable CD1.0 or CD1.1 transmitters, scroll down to *Data transfer* and select *CD1.0—Transmitter* or *CD1.1—Transmitter*. Fill in the required data fields and click *Save settings*. The meaning of each field is explained on the page and in [Chapter 4, “Configuration options”](#).

Using an Authentication Module (AM) you can cryptographically sign all outgoing CD1.0 or CD1.1 subframes as they pass through. This is particularly powerful where the module is physically part of the sensor casing, as it allows you to generate authenticated data at source, even within a borehole.

Optionally, the DCM can also accept CD1.0 or CD1.1 data from another instrument. Incoming data is stored in Flash memory, just like that coming from attached sensors, and can be retransmitted as CD1.0, CD1.1, GCF, or by any other method as required.

AutoDRM

AutoDRM (Automatic Data Request Manager) is an optional service which allows you to request data over e-mail. In response to a request, the AutoDRM can either send an e-mail in return, or establish an FTP connection to you.

In order for the AutoDRM system to work, the module must be able to send and receive e-mail messages. To set the system up, you will need to fill in the details of your network's SMTP server under *Outgoing mail setup*, and of your POP or IMAP server under *Incoming mail setup*. Remember to click *Save changes* after you have filled in each screen.

You can now enable the AutoDRM system under *Data transfer—AutoDRM*. Click *Save changes* to start the service.

To test the AutoDRM system, send the following e-mail to the DCM:

```
From: your@e-mail.address  
To: autodrm@your.DCM
```

```
BEGIN GSE2.0  
MSG_TYPE request  
MSG_ID unique-identifier  
HELP  
EMAIL your@e-mail.address  
STOP
```

where your@e-mail.address is the e-mail address to send the return message and [unique-identifier](#) is a string which you can use to identify your request. autodrm@your.DCM should be replaced with an e-mail address which will be delivered to the `autodrm` mailbox on the DCM.

The DCM should send a mail back to you containing help on the commands you can give the AutoDRM.

3.5 Troubleshooting DCM installations

I cannot connect to the DCM's DATA OUT port using a terminal program.

Press `ENTER` a few times to initiate communication.

If the *DATA OUT* port gives a `login` prompt, you should log in with your username and password. If you have not been given a different username and password, try logging in as `root` with the password `rootme`. You should change this password as soon as you can with the command `passwd`

If the *DATA OUT* port gives an `ok` prompt, the DCM is running the *gcf_out* service on that port and has provided you with a FORTH compatibility interface. Type

```
GETTY
```

and press `ENTER` to gain access to the login prompt.

If the *DATA OUT* port does not respond when you press `ENTER`, or produces garbage, check that your terminal program is using the same baud rate as the DCM. By default, the DCM uses a baud rate of 115200, with 8 data bits, no parity bit and 1 stop bit, and no flow control.

If the *DATA OUT* port is not responding at all, check the power supply to the DCM.

I cannot see the DCM's Web site over HTTP or HTTPS.

There is a problem with the network setup. Connect to the DCM's *DATA OUT* port using a terminal program and change the network settings to suit your network. In particular:

- If the DCM has a static IP address, use `ifconfig` to verify that the DCM is using the correct IP address. If it is not, change it with

```
gcf gdbset net.eth.0 static
gcf gdbset net.eth.0.netmask 255.255.255.0
gcf gdbset net.eth.0.address 192.168.0.2
```

(replacing 192.168.0.2 with the IP address you want the DCM to use.)

- If the DCM is using DHCP, use `ifconfig` to verify that the DCM is using the IP address you expected. If it is not, change the settings of your DHCP server or connect to the correct IP address for the DCM
- Check that your local PC can route to the DCM's IP address. For example, if you are using a cross-over Ethernet cable, the two hosts must share a subnet.
- Check that the DCM's HTTP(S) server is enabled by issuing one of

```
gcf gdbset net.remoteaccess.allow.http yes
gcf gdbset net.remoteaccess.allow.https yes
```

as necessary.

I cannot connect to the DCM's Scream! server.

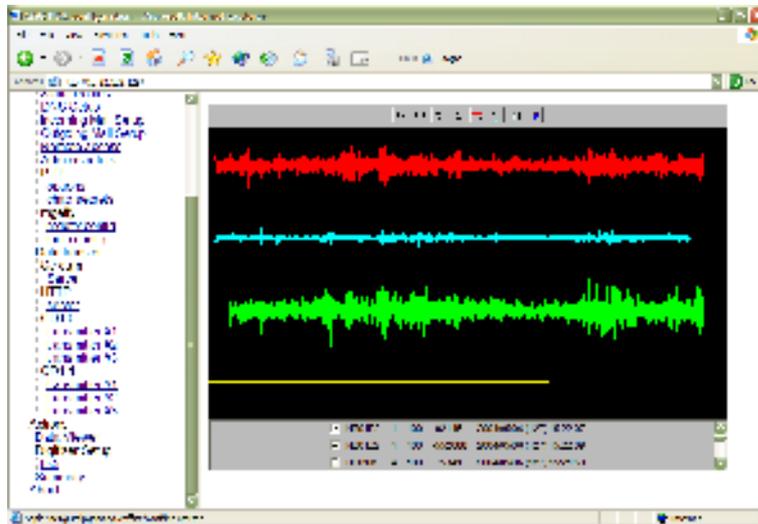
Make sure you are using UDP to connect to the DCM. Scream! servers between computers (including the DCM) always use UDP. The **Add TCP server...** option in Scream! is intended for hardware serial-to-IP converters only.

I cannot GCFPING the DCM's Scream! server.

- Check the **Receive UDP data** checkbox in Scream!'s *Network Control* window.
- Open the DCM's Web site, and check that the option `datatransfer.scream.server` is set to *on*, and that `datatransfer.scream.server.port` is the port you expected.

I can GCFPING the DCM's Scream! server, but no data appears.

Check the DCM is receiving data by selecting **Actions** → **Data Viewer** on its Web site. This is a Java applet which provides some of the functionality of Scream!, allowing you to check that data is being received correctly.



The streams being received at the DCM are listed in the bottom section of the applet. Click on a checkbox to add that stream to the main viewer window.

If you do not want to use the DCM's Web site, you can find out the number of GCF blocks the DCM has received with the command `gnblocks`. Each serial port will be listed, with its name, number, key number and device name (as for `serialmap`) but including the number of blocks received on that port:

```
Key 0x007005: Blocks      3287 (Port  5, name Port A (COM6),
                                device /dev/ttyS2, baud 9600)
```

The DCM is not receiving any data streams.

- Check the connection between the digitizer and the DCM by trying to log in to the digitizer's console using any of the methods described in [“Communicating with digitizers” in Section 3.4](#). Press ENTER a few times to initiate communication.
- If the digitizer gives an `ok` prompt when you press ENTER, check that you have configured the digitizer to output real time data streams. Streams will not appear until a whole GCF block (1024 bytes) is ready for transmission, so a 5 sps stream may not appear until the digitizer has been working for 4 minutes. In addition, you can configure a digitizer to output only triggered streams, in which case it will not appear until a trigger occurs. (There is an exception to this: if you have put the

digitizer in the *FILING* or *DUAL* filing mode, it will send heartbeat messages to Scream! clients every so often. The DCM will not show these messages in the *Data Viewer*.)

- If the digitizer does not respond when you press ENTER, or produces garbage, check that the DCM is using the same baud rate as the digitizer. By default, digitizers use a baud rate of 9600, with 8 data bits, no parity bit and 1 stop bit, and no flow control. To change the DCM's settings, *exit the terminal program*, and either
- use the **Serial port configuration** → *port name* Web page, or
- issue configuration commands such as

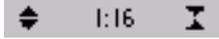
```
serial.5.service gcf_in
serial.5.baudrate 9600
serial.5.handshaking none
```

To obtain the port number (here 5) corresponding to a named port, issue the command `serialmap`.

- If the digitizer is not responding at all, check the power supply to the digitizer and the cable between it and the DCM.

The DCM is receiving streams, but they do not contain any data.

If the data you see is zero:

- Increase the Y axis scale using the  icon in Scream!'s *Waveview window* or the corresponding icon in the DCM's *Data Viewer*.
- Check the mass position outputs to see if the masses are locked. A properly unlocked and centred mass should show a mass position within 1000 counts of zero; a locked mass will give a value in the region of 32,000 or -32,000 counts. If your sensor has remote mass locking, you can unlock the masses by navigating to **Actions** → **Digitizer setup** and clicking *Unlock sensor*.

Locking or unlocking the sensor mass typically takes several minutes to complete.

- Check the connection between the sensor and digitizer and try again.

If you cannot see the data in the stream, remove any constant offsets by clicking on the **Zero streams** icon  in Scream!'s *Waveview window* or the DCM's *Data Viewer*.

The DCM receives streams, but gaps appear in the data some minutes after boot-up.

Check that the baud rate between the digitizer and the DCM is sufficient for all the data streams you want to transmit. If it is not, the digitizer's output buffer will gradually fill up until no more data can be stored. Increase the baud rate of the digitizer through the DCM (or using `Scream!`), *then* set the baud rate of the DCM's input port to the same value.

If you are using triggered output streams, be especially careful to allow a high enough baud rate to transmit data from all possible output streams simultaneously, or you will observe gaps when an event triggers the digitizer.

The DCM receives streams, but 2-minute gaps appear in the data at 4-hour intervals.

After a reboot, the DCM takes around 2 minutes to begin transmitting. The DCM runs a `guardian` process which monitors the health of the system. In some circumstances `guardian` will need to reboot the DCM to attempt to resolve a problem. If the reboot does not help, the DCM will soon find itself in the same position, and `guardian` will reboot it again.

You can check the time since the last reboot with the command `uptime`, which will respond with a line like

```
14:30:32 up 34 min, load average: 1.24, 1.32, 1.10
```

In this example, the DCM last rebooted 34 minutes ago.

The most common circumstance where `guardian` will reboot a DCM is when the operator has instructed it to record data, but it cannot do so, either because

- the Flash memory is full, and all connected USB disks are also full,
- the Flash memory is full, and no USB disk is present; or
- the Flash memory is full, and a USB disk is inaccessible for some other reason (*e.g.* it is unformatted, incorrectly partitioned, or faulty).

4 Configuration options

All DCM modules feature an on-board Web server, which you can use to set up the unit. When you first connect to the module, you will be shown a menu tree in a panel on the left, with a summary of its current status (see [Section 5.7, “Summary”](#)) on the right.

Links in the section of the menu tree marked *Configuration* allow you to change the settings used by the DCM itself. Once you have changed settings on any page, you should click **Save changes** at the bottom of the page for the DCM to make your new choices active. If you move to a different page without saving the settings, your changes will be lost.

To change the configuration of the digitizer(s) and instrument(s) attached to the DCM, you should use the **Actions** → **Digitizer Setup** page (see [Chapter 5, “Actions”](#).)

Note: The DCM is a highly flexible unit, which can be supplied with a range of feature sets depending on your requirements. A stand-alone DCM unit normally has most of these options available to the user. However, some of the settings described below may not be relevant to your installation, and thus will not be available.

Conversely, a DCM may be supplied which supports multiple instances of a particular service (*e.g.* several CD1.1 transmitters.) In this case, each instance will have its own configuration page, which will appear as described below for that service. Users of installations which include several types of DCM or AM (Authentication Module) will find that each unit offers a different range of configuration options, according to its specialization as a component in the system.

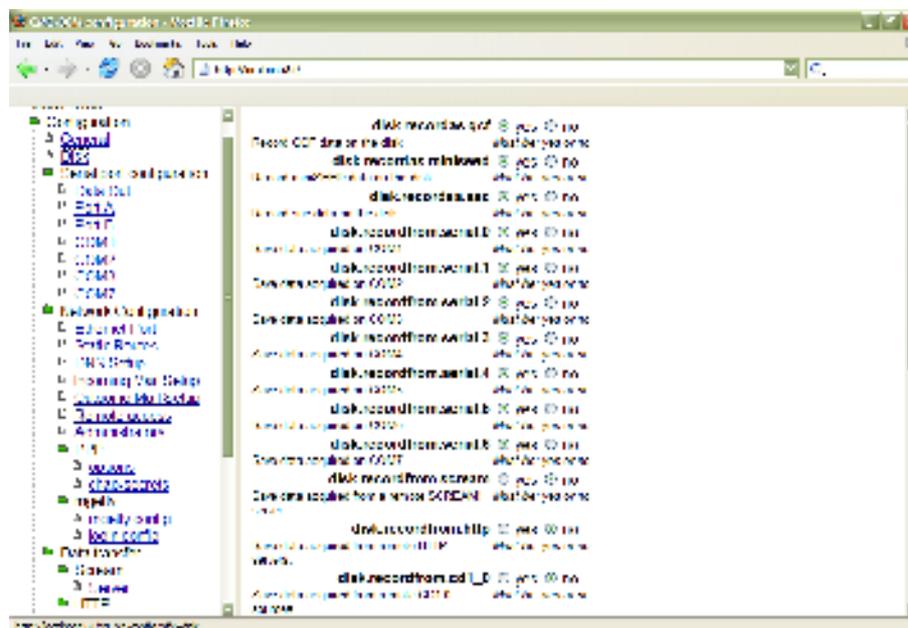
In this chapter, the name of each setting is marked in *italics*. This name is shown on the Web page interface, and can also be used to access the setting using the `gcfgdbset` command (see [Section 6.2, “Command line tools”](#).)

4.1 General

This page contains a single setting.

general.hostname : This is the host name the DCM will use to identify itself. It is normally the first part of the module's fully-qualified domain name, for example, a DCM at *myinst.bignet.org* would have its host name set to *myinst*.

4.2 Disk



This page allows you to change how the DCM will use its hard disk, and what data it will record on it. If you want to check that the hard disk is working, explore its contents, or format it, you should use the *Disk tools* page instead (see below.)

The first three options allow you to specify formats in which the DCM should record data. There are three options: *gcf*, *miniSEED* and *sac*. The latter two are provided for archival purposes, and are not optimised for data transfer; Güralp digitizers and software all work natively with GCF, the Güralp Compressed Format. The DCM allows you to record in several formats simultaneously, if you wish; you should make sure that you have the capacity to store, or the bandwidth to transfer, all the data the DCM will produce in each format you have enabled.

One file will be prepared for each format (or format and stream, for those formats which deal with single streams) every three hours. The DCM will always start a new file if it is power cycled; the aborted file will still be valid. Files are not immediately written to the disk, but into Flash memory; the options *disk.usagemode* and *disk.writeinterval*, below, control how often the DCM flushes new data to the disk, and what it will do if the disk becomes full.

disk.recordas.gcf : Set this to *yes* to have the DCM record Güralp Compressed Format data files.

disk.recordas.miniseed : Set this to *yes* to have the DCM record files in miniSEED format.

disk.recordas.seed : Set this to *yes* to have the DCM record full SEED volumes on the hard disk. This option will have no effect until you create a *datatransfer.seed.config* file. See [“SEED” in Section 4.5](#) for information on these options.

disk.recordas.sac : Set this to *yes* to have the DCM record files in *sac* format.

The next set of options allows you to select the data sources for archival.

disk.recordfrom.serial.n : Each of these options refers to a serial port on the DCM by its number. Select *yes* to have the DCM record data coming into this port.

disk.recordfrom.scream : Select *yes* to have the DCM record data it receives from remote Scream! servers (either computers running Scream!, or further DCMs and AMs.) If you want to use this option, you should also configure the *datatransfer.scream.client* options (see below.)

disk.recordfrom.http : Select *yes* to have the DCM record data it receives using the HTTP protocol. If you want to use this option, you should also configure the *datatransfer.http.client* options (see below.)

disk.recordfrom.cd1_0 : Select *yes* to have the DCM record data it receives using the CD1.0 protocol. If you want to use this option, you should also configure the *datatransfer.cd1_0* options (see below.)

disk.recordfrom.cd1_1 : Select *yes* to have the DCM record data it receives using the CD1.1 protocol. If you want to use this option, you should also configure the *datatransfer.cd1_1* options (see below.)

disk.power : To save power, the on-board hard disk can be set to power up only when required. If power consumption is an issue and you expect the hard disk to be used only occasionally, you should set this option to *automatic*. Otherwise, setting it to *always-on* will ensure that the hard disk is always ready to receive data. If you intend to operate an internal disk continuously, you should take care not to let it overheat, since the interior of the DCM is well insulated.

disk.usagemode : This setting varies the way the DCM uses its hard disk storage once it has been filled up.

- The *use-once* option causes the DCM to stop recording data to the hard disk, and discard any new data that arrives. It may still send new data over the network, if you have so configured it, but the hard disk will be untouched once full. Thus the *start* point of the stored data is known, but the end point depends on the disk capacity. This is most useful for installations where you are expecting to replace the hard disk periodically, or where you need to take time-synchronized readings at several sites.

The stand-alone DCM has three RS232 serial ports brought out on 10-pin mil-spec connectors, labelled *PORT A*, *PORT B* and *DATA OUT*. The labelling is provided for convenience, and can be ignored if it does not match your site requirements: each port may be used for communication in either or both directions.

Each serial port has its own configuration screen, with a number of options beginning *serial.x* (where *x* is the internal number of the serial port). The configuration options for all the ports are as follows:

serial.x.baudrate : This option alters the speed of communication across each serial link, in bits per second.

For ports connected to digitizers, you should ensure that the baud rate is high enough to allow all the data to be transmitted at the rates you have chosen. As an example, for three streams transmitting at 100 Hz, a rate of 9600 baud is usually sufficient. Modern modems can normally operate at rates up to 57600 baud (~56 kbits/s), although the telephone or transmission lines may not support such a high rate. The same is true of radio telemetry links.

The DCM's serial ports operate using frames of 8 data bits, no parity bits, and one stop bit.

serial.x.device : The pathname of the Linux device corresponding to this serial port. You should not need to change this option.

serial.x.handshaking : The flow-control (handshaking) protocol used across each serial link. There are three options:

- *off* : Transmit data across the serial link without handshaking, *i.e.* assume that the link is always ready to send or receive data.
- *rts/cts* : Use the Ready To Send/Clear To Send handshaking method, where two separate lines within the serial cable are used to control the flow of data. This is the most reliable method since accomplished in hardware, but is not feasible for long-distance or complex connections.
- *xon/xoff* : Send the two special characters ^Q (17) and ^S (19) as part of the data stream to request that data transfer be started and stopped. This method requires only a single connection, but assumes that the special characters will be received correctly every time.

serial.x.name : The name of this serial port. You should not ordinarily change this setting.

serial.x.ppp : You can use any of the serial ports to establish a network connection with a computer using the PPP protocol by setting the *serial.x.service* option, below, to *ppp*. The DCM uses the standard Linux

command `pppd` to establish these links. `pppd` is a versatile program, which uses command-line options to specify the type of PPP link required. These options are fully described in the Linux documentation for `pppd`.

The *serial.x.ppp* box allows you to specify `pppd` options to use *on this serial port*. Options which you wish to apply to all serial ports running the *ppp* service should be placed in the `/etc/ppp/options` file (see [“options” in Section 4.4.](#))

serial.x.service : How the DCM will use each serial port. There are five options:

- *getty* : The module will use this port to listen for logins from a connected computer. Once logged in, the user will have direct command-line access to the DCM's Linux operating system.
- *mgetty* : The module will use this port to listen for logins over a connected serial modem. Again, a logged-in user will be presented with the DCM's Linux command line.
- *mgetty-r* : The module will use this port to listen for logins over a connected serial modem which is operating in raw mode (*i.e.* which does not support the standard *AT* modem commands.) This option runs the Linux command `mgetty -r`. A port running the *mgetty-r* service will also listen for incoming PPP frames, and if it detects one will configure the PPP link according to the `/etc/mgetty+sendfax/login.config` file (see [“login.config” in Section 4.4.](#))
- *ppp* : The module will use this port to provide a PPP link to a connected computer. Once set up on the computer, the DCM will appear as if on a local TCP/IP network. You can then log in to it using *ssh* or use its Web-based administration system, as well as receiving data.

Before a PPP link will function, you will need to specify the PPP options and (if using CHAP) provide a `chap-secrets` file. This can be done through the DCM configuration interface: see [“options” in Section 4.4](#) and [“chap-secrets” in Section 4.4](#). Also see the Linux manual page for `pppd` (8) for an explanation of how PPP is implemented.

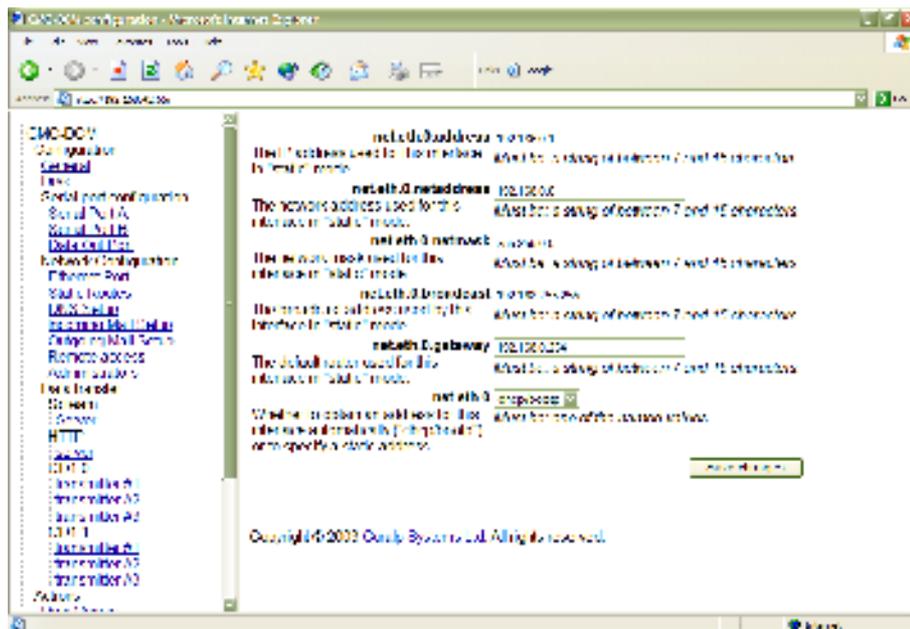
- *gcf_in* : The module will use this port to communicate with a GCF-compatible digitizer.
- *gcf_out* : The module will use this port to send GCF data directly to the connected device as it is received. A number of programs are available for PCs which handle incoming GCF data streams, such as *Scream!*, *Antelope*, and *Earthworm*. The DCM transparently merges all incoming data streams, so if you are using *Scream!*, you can also configure and

control attached digitizers using its own interface.

4.4 Network configuration

These settings must be provided if you intend to connect a DCM to a TCP/IP network, whether over a serial link via PPP, or using Ethernet or wireless connections.

Ethernet port



The settings described below begin *net.eth.0* because they refer to the first Ethernet interface. If the optional 802.11b (“Wi-Fi”) hardware is installed, the settings on this page will begin *net.wlan.0* and refer to the wireless LAN rather than to Ethernet networking. They are otherwise identical.

net.eth.0 : If you know the IP address the DCM is going to use over this interface, you should set this option to *static* and fill in the details below. Alternatively, you can select *dhcp/bootp* and have the DCM automatically request an address over the network. If you choose the *dhcp/bootp* option, you should ensure that a gateway computer exists on your network that can receive these requests, and that the DNS (nameserver) entry for the DCM is kept current. You can then ignore the remaining settings on this page.

net.eth.0.address : If the above option is set to *static*, you should fill in the IP address of the interface in this box. Valid IP addresses consist of four numbers between 0 and 255, separated by periods. If you intend to attach the DCM to a private network, you should use an IP address in one of the private ranges (10.b.c.d; 172.16-31.c.d; 192.168.c.d) to avoid clashing with addresses on the wider Internet.

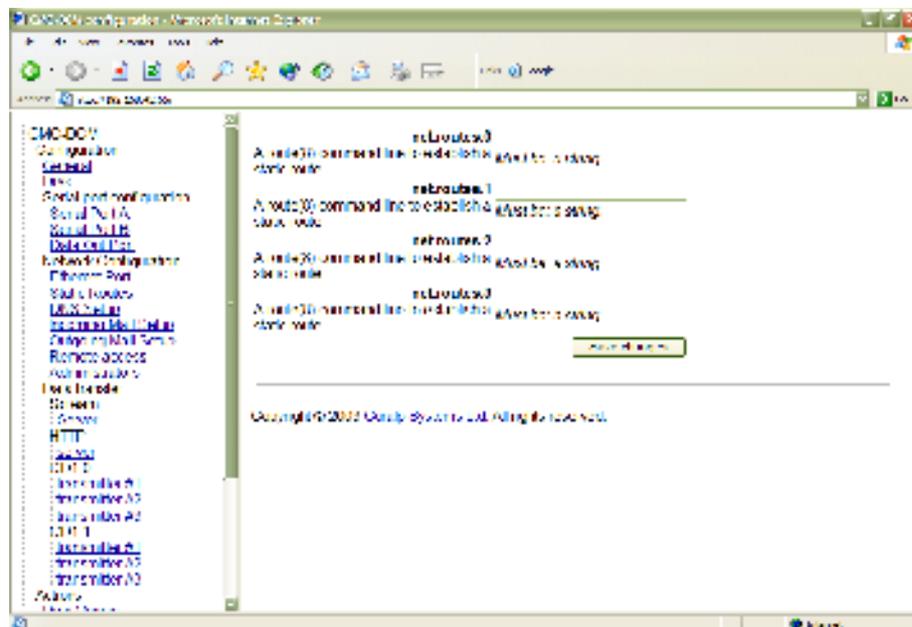
net.eth.0.broadcast : The broadcast IP address used by the interface. This can often be derived from the IP address by replacing numbers at the end with 255.

net.eth.0.gateway : The IP address of the gateway machine (router) on your network.

net.eth.0.netmask : The network mask to use for the IP interface. A DCM with an IP address in the domain 192.168.1.x would normally use a netmask of 255.255.255.0.

net.eth.0.netaddress : The network address of the DCM. A DCM with an IP address in the domain 192.168.1.x would normally use a network address of 192.168.1.0.

Static routes



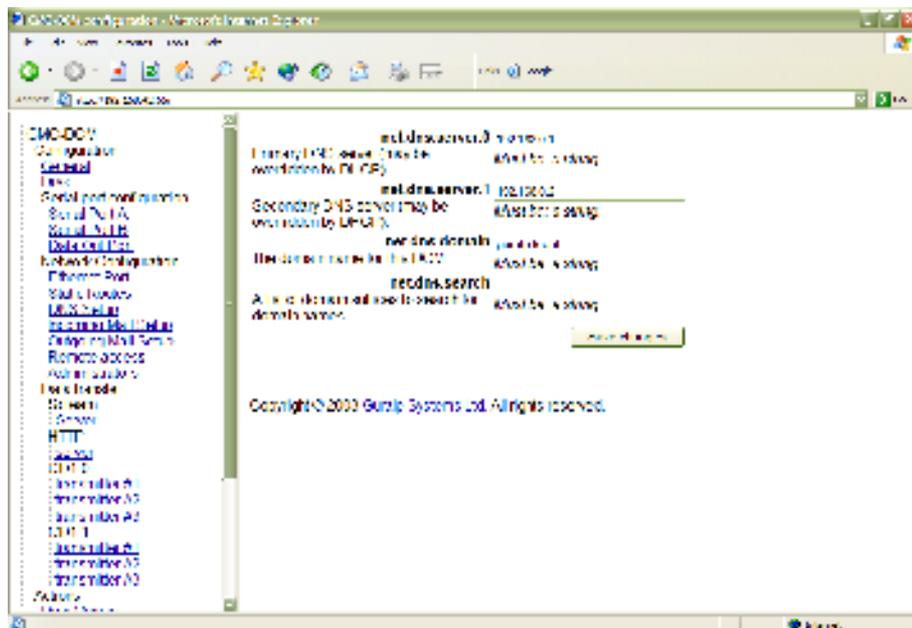
In addition to the standard IP routing mechanisms, you can define up to 4 additional static routes from the DCM to other parts of the network. These routes are specified as arguments to the standard Unix `route` command. For example, if you wanted the DCM to route traffic to and from IP addresses beginning 192.168.0. over the `eth0` interface, you would include the line

```
add -net 192.168.0.0 netmask 255.255.255.0 dev eth0
```

in the list.

For more information on the `route` command, please see its Linux manual page.

DNS setup



net.dns.domain : The name of the domain containing the module. For example, a DCM at *myinst.bignet.org* would have its domain set to *bignet.org*.

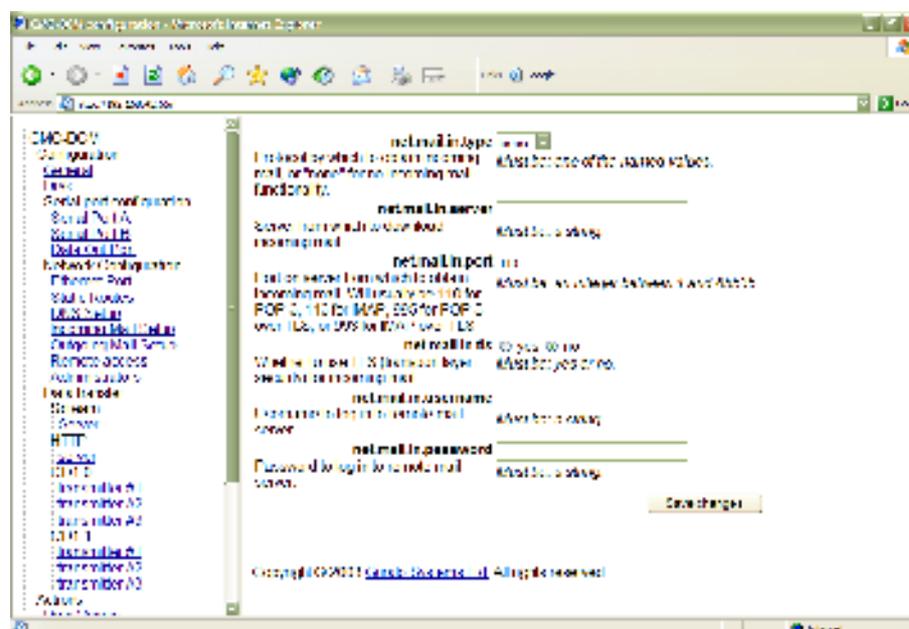
The rest of the settings on this page determine how the DCM determines the addresses of other hosts on the network from their host names.

net.dns.search : If the DCM encounters an unqualified hostname in any of its configuration settings, or in a command, it will look for that name in each of the domains specified here in turn. You should give the domains to search in their correct order, separated by commas. The DCM's own domain (as given in the *net.dns.domain* setting) will always be searched first, so you need not specify it here.

net.dns.server.0 : The IP address of the primary nameserver used to resolve host names. If you are using *dhcp/bootp*, the DNS service is negotiated automatically, and so this setting may be ignored.

net.dns.server.1 : The IP address of the secondary nameserver. Again, you may be able to ignore this setting if you are using *dhcp/bootp*.

Incoming mail setup



This page enables you to set up the DCM to retrieve e-mail from a POP3 or IMAP server on the network, either unencrypted or secured using TLS (transport-layer security.) If you do not need the DCM to be able to fetch mail, select *None* from the *net.mail.in.type* drop-down menu and ignore the remaining settings.

net.mail.in.password : The password needed to log in to the mail server.

net.mail.in.port : The number of the port to connect to on the mail server. This generally depends on the protocol used to retrieve mail:

- For *pop3* mail, this port will normally be 110, or 995 for secure POP3 over TLS.
- For *imap* mail, this port will normally be 143, or 993 for secure IMAP over TLS.

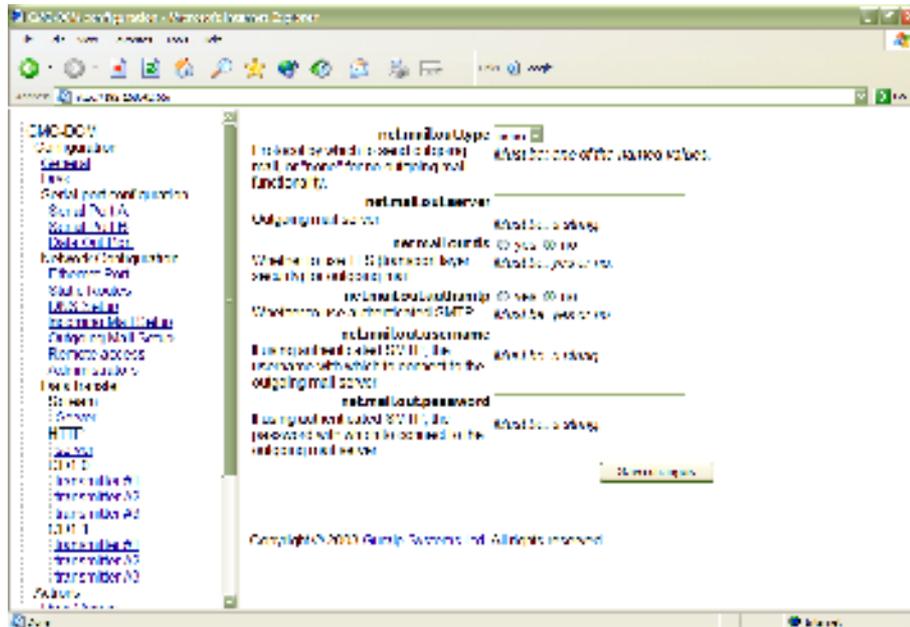
net.mail.in.server : The hostname or IP address of the mail server.

net.mail.in.tls : Select *yes* to encrypt the incoming mail channel with TLS, *no* to leave communications unencrypted.

net.mail.in.type : Select *pop-3* or *imap* to use that protocol to retrieve mail from the server. If you do not need the DCM to be able to receive mail, select *None*.

net.mail.in.username : The username needed to log in to the mail server.

Outgoing mail setup



This page enables you to send mail from the DCM using the SMTP protocol. Both authenticated (*i.e.* sending a username to the server) and unauthenticated SMTP are supported, as is encryption using TLS (transport-layer security.) If you do not need the DCM to be able to send mail, select *None* from the *net.mail.out.type* drop-down menu and ignore the remaining settings.

net.mail.out.authsmtp : Select *yes* and give a username and password in the fields below to use authenticated SMTP for outgoing mail.

net.mail.out.password : The password needed to log in to the outgoing mail server, if using authenticated SMTP.

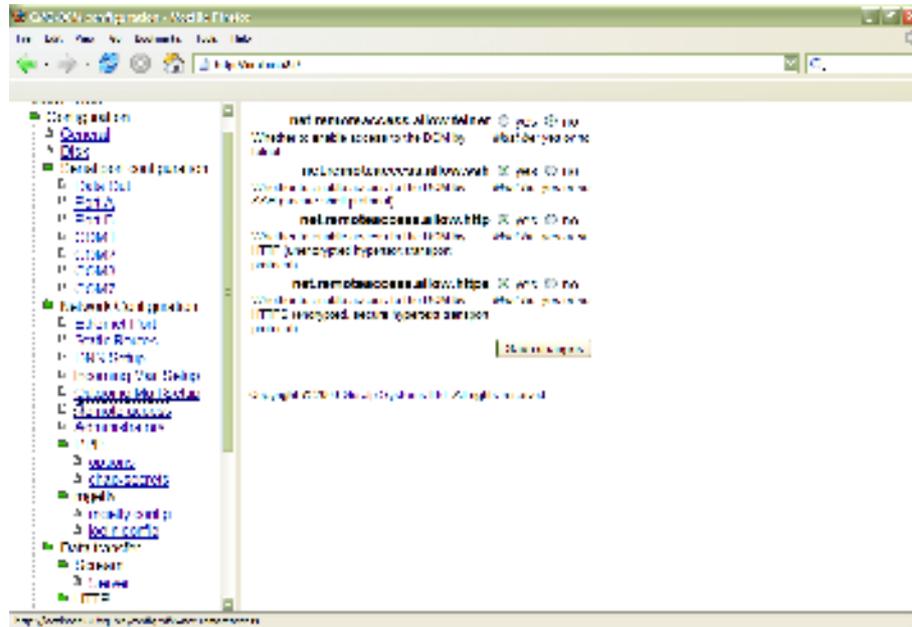
net.mail.out.server : The hostname or IP address of the outgoing mail server.

net.mail.out.tls : Select *yes* to encrypt the outgoing mail channel with TLS, *no* to leave communications unencrypted.

net.mail.out.type : Select *smtp* to use that protocol to send mail to the server. Currently only SMTP is supported. If you do not need the DCM to be able to send mail, select *None*.

net.mail.out.username : The username needed to log in to the outgoing mail server, if using authenticated SMTP.

Remote access



The settings on this page determines which ways you can use to access the DCM over a TCP/IP network. If all these methods are disabled, you may still be able to connect to the DCM over a serial link using *getty* (see *serial.x.service*, above), or over a direct modem connection. Your settings will come into effect as soon as you click *Save settings*, so you should ensure that you have a backup means of communication with the DCM before disabling these methods.

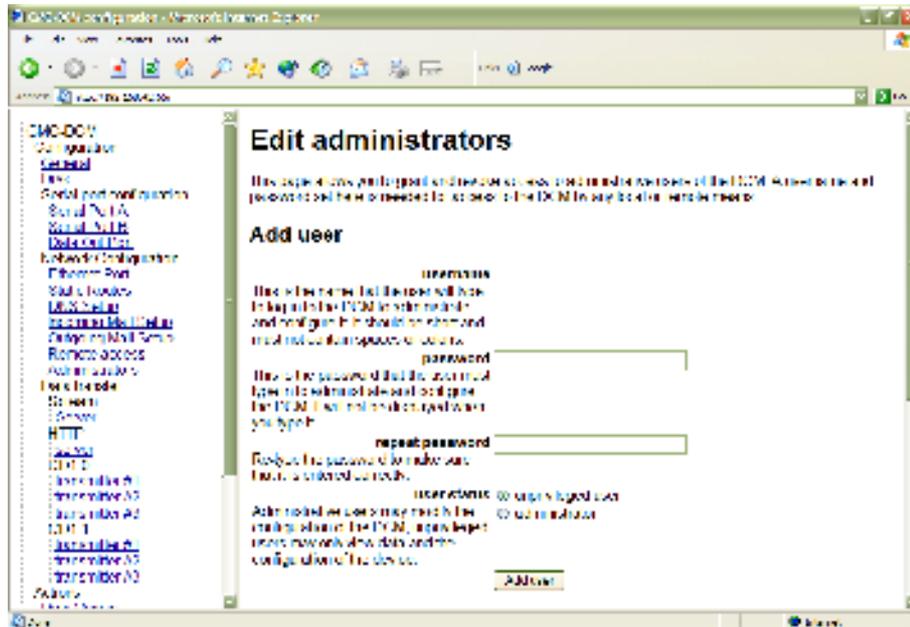
net.remoteaccess.allow.telnet : Select *yes* to allow telnet clients to log in to the DCM's Linux operating system, or *no* to disallow connections over telnet. Select *no* to disable it. By default, this method is disabled for security reasons.

net.remoteaccess.allow.ssh : Select *yes* to allow SSH clients to log in to the DCM's Linux operating system, or *no* to disallow connections over SSH. By default, this method is enabled.

net.remoteaccess.allow.http : Select *yes* to allow access to the Web-based administration system by unencrypted HTTP. (A username and password will still be required, but they are sent over the network insecurely.) Select *no* to disable it. By default, this method is disabled for security reasons.

net.remoteaccess.allow.https : Select *yes* to allow access to the Web-based administration system by HTTPS (secure HTTP), or *no* to disable it. By default, this method is enabled.

Administrators



Users of the DCM can be one of two kinds.

- *Unprivileged users* can log in or view the Web interface of the DCM, including its current configuration, but cannot change any of the options, nor configure any instruments attached to the DCM
- *Administrators* may change settings or configure attached digitizers as they please.

You may want to give different users separate accounts for tracking or other purposes, or to allow users to set their own preferences when logging in. Alternatively, you may prefer to restrict access only to those with administrative requirements, by removing any unprivileged user accounts. If you want to access the DCM by *any* means, locally or remotely, you must have a username and password on the list given here.

If this is the first time you have logged in to the DCM, there will be a single user, `root`, whose password is factory-set. To add a new user, fill in their username and password in the top two boxes of the page. Repeat the password in the next box, to ensure it is entered correctly (since it will not be visible on screen), and click on **Add user**. A username may not contain spaces or colons.

You should be shown an acknowledgement screen, indicating that the new user account has been created. The user will be able to log in with that username and password by any of the methods currently available to the device, and from any location. More complex access controls cannot be performed through the Web administration system, although you can of course use the features of the DCM's underlying Linux operating system to implement them if required.

To remove a user account, choose their username from the drop-down box below *Existing accounts* and click **Delete user**. If you delete your own account, you will no longer be able to log in to the DCM, although you may finish what you are doing in the current session.

You can also change the password for any account, for example if you believe it to have been compromised, or if you are assigning it to a new person. To do this, choose their username from the same drop-down box and repeat the new password in the two boxes at the bottom of the page. Again, the password will not be visible on screen. Click *Change password* to make the change.

PPP

This submenu allows you to alter the PPP configuration files, `/etc/ppp/options` and `/etc/ppp/chap-secrets`. These two files are required by the Linux PPP server, and affect all *ppp* services on the DCM. (Separate options for each port may be specified under *Serial port configuration*; see [Section 4.3, “Serial port configuration”](#)). The `/etc/ppp/chap-secrets` file is only necessary if you are using CHAP to authenticate PPP connections.

Clicking on either of these entries brings up a page in the work area which enables you to edit the file and its attributes directly, including

- the ownership of the file;
- the permissions (read, write, or execute) of the file for its own user, its own group, and all others; and
- the content of the file.

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, referring to the Linux manual page for `pppd` (8) if you are unsure.

When you have finished editing a file, clicking **Save changes** will write the changes to disk. The changes will not take effect, however, until the PPP services are restarted. You can restart all running PPP services by clicking *Restart PPP*; whilst this occurs, the network will be briefly unavailable.

options

The file `/etc/ppp/options` sets the default options for the PPP daemon. These will be applied to all PPP connections and to any of the serial ports which are running the *ppp* service. The file is treated as a list of words, each either an option or an argument to a previous option. For a list of the available options, and a full explanation of the format of the `/etc/ppp/options` file, see the Linux manual page for `pppd` (8).

You can specify additional options for each serial port separately using the *serial.x.ppp* configuration option, which can be found on the *Serial port configuration* page for that port.

chap-secrets

The file `/etc/ppp/chap-secrets` contains secrets for `pppd` to use when it authenticates itself to other systems, and also when it authenticates other systems to itself. Each line in a secrets file contains (at least) the name of a client, the name of the server, and a secret specific to that particular combination of client and server.

For a full explanation of the format and usage of the `/etc/ppp/chap-secrets` file, see the Linux manual page for `pppd(8)`.

mgetty

This submenu allows you to alter the *mgetty* configuration files, `/etc/mgetty+sendfax/mgetty.config` and `/etc/mgetty+sendfax/login.config`.

Clicking on either of these entries brings up a page in the work area 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, referring to the Linux manual page for `mgetty(8)` if you are unsure.

When you have finished editing a file, clicking **Save changes** will write the changes to disk. The changes will not take effect, however, until all *mgetty* services are restarted. You can restart all running *mgetty* services by clicking *Restart mgetty*; whilst this occurs, the network will be briefly unavailable.

mgetty.config

The file `/etc/mgetty+sendfax/mgetty.config` is the main configuration file for *mgetty*.

For full details on the format of the file and the options available, see the explanatory comments within the file, or the Linux info documentation for `mgetty`.

login.config

Normal user logins are handled by the program `/bin/login`. However, you may want to run a different program to handle logins by certain users. You can do this by editing the file `/etc/mgetty+sendfax/login.config`.

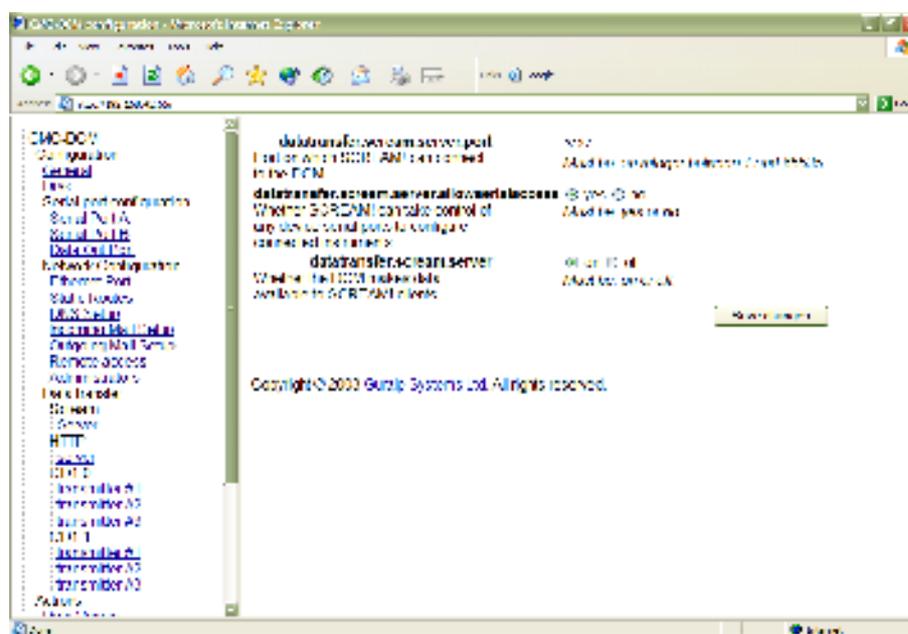
For full details on the format of the file, see the explanatory comments within

the file, or the Linux info documentation for `mgetty`.

4.5 Data transfer

You may retrieve data from the DCM using a variety of standard seismic protocols and formats. In addition, the DCM includes facilities for communicating with Güralp Systems' own Scream! PC software. The options on this set of pages allow you to configure the DCM to use these additional methods.

Scream!



Scream! is a Microsoft Windows software package designed to configure and retrieve data from Güralp digitizers. The options on this page allow you to control how the DCM responds to requests from Scream!. Depending on your requirements, the DCM can act either as an instrument in itself, or pass through requests directly to connected equipment.

`datatransfer.scream.server` : Select *on* to allow Scream! to connect to the module over a TCP/IP connection and receive data as if it were a connected instrument. The connection can be established over an Ethernet, wireless, or PPP link. If you do not want to transfer data directly to Scream! clients, select *off*.

`datatransfer.scream.server.allowserialaccess` : The Scream! software allows you to configure digitizers connected to it by serial links. Select *yes* if you want to be able to use Scream! to configure digitizers attached to the DCM's serial ports. The software is able to differentiate between several instruments connected to the DCM, so enabling this option will allow you to configure all attached digitizers from within Scream!.

datatransfer.scream.server.port : The network port number which Scream! should use to connect to the DCM. You can use any port which neither the PC nor the DCM is using for other purposes. See your Scream! configuration for the current port setting; the default is 1567.

datatransfer.scream.sampleratelimit : The fastest sample rate allowed for transmission over this connection. Using this option, you can prevent high-rate streams from being transmitted to Scream!, whilst allowing them to be transmitted by other means (e.g. CD1.1, etc.) Setting this option to zero will filter out all data streams but preserve status channels, allowing you to use Scream! to monitor the status of your instrumentation whilst minimizing the bandwidth used.

Scream! Client

datatransfer.scream.client : The DCM also has the ability to act as a Scream! client, *i.e.* to retrieve data from another Scream!-compatible instrument or server on your network. This data is treated as if it came directly from an attached digitizer, and will be collected on the hard disk or sent out across the network according to the module's configuration. If you wish to be able to do this, select *on*.

datatransfer.scream.client.port : The network port number which the DCM should use to connect to the remote Scream! server. You can use any port which neither device is using for other purposes. The default is 1567.

datatransfer.scream.client.server : The IP address or hostname of the Scream! server from which you wish the DCM to gather data.

AutoDRM

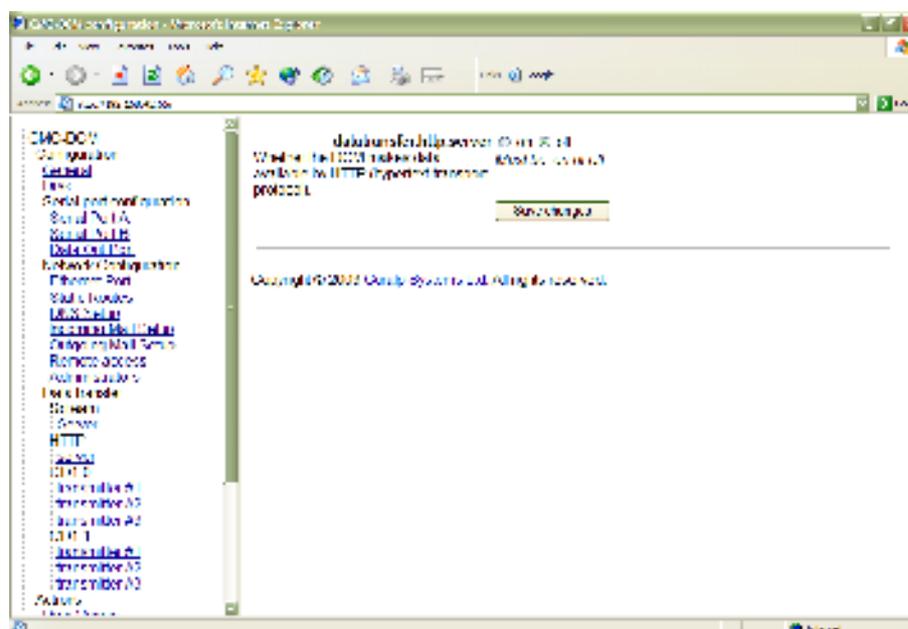
AutoDRM is an automated system for handling requests for seismic data over e-mail, and for fulfilling those requests either by returning e-mails or by establishing an FTP connection. In order for the AutoDRM system to work, the module must be able to send and receive e-mail messages: this can be configured using the *Incoming* and *Outgoing mail setup* configuration pages. See [“Incoming mail setup” in Section 4.4](#) for more details.

The DCM's AutoDRM capabilities are compatible with the GSE2.0 and 2.1 message standards developed for GSE-EISMS and GSETT-3.

datatransfer.autodrm : Select *on* to enable the DCM to handle AutoDRM requests.

datatransfer.autodrm.maxpoolsize : The maximum pool size to use whilst processing AutoDRM requests, in megabytes.

HTTP server



In addition to configuring the DCM and instruments attached to it, you can also instruct the on-board Web server to provide data to network clients.

datatransfer.http.server : Select *on* if you wish to be able to retrieve data from the DCM over the Web (*i.e.* HTTP). Once the HTTP server is enabled, you can browse through the DCM's hard disk using the URL <http://mydcm/cgi-bin/exploreifs?path=/>.

You may need a username and password to access this service.

HTTP client

The DCM also has the ability to retrieve data from another instrument or server on your network over HTTP.

datatransfer.http.client : Select *on* if you wish the DCM to retrieve data from networked instruments over HTTP.

datatransfer.http.client.url : The URL from which the DCM will collect data by HTTP.

CD1.0

Continuous Data Format, CD1.0, is a standard method for sending seismic data over a TCP/IP network. The DCM module can send and receive data in CD1.0 format to a specific CD1.0-compatible device or NDC.

Some installations of the DCM have multiple CD1.0 senders. In this case, there

will be several *sender* pages, all with the same options.

datatransfer.cd1.sender : Select *on* to activate the CD1.0 sender. If you do not want the DCM to transmit CD1.0 data, select *off* and ignore the remaining settings.

datatransfer.cd1.sender.sitename : The CD1.0 site name for the DCM.

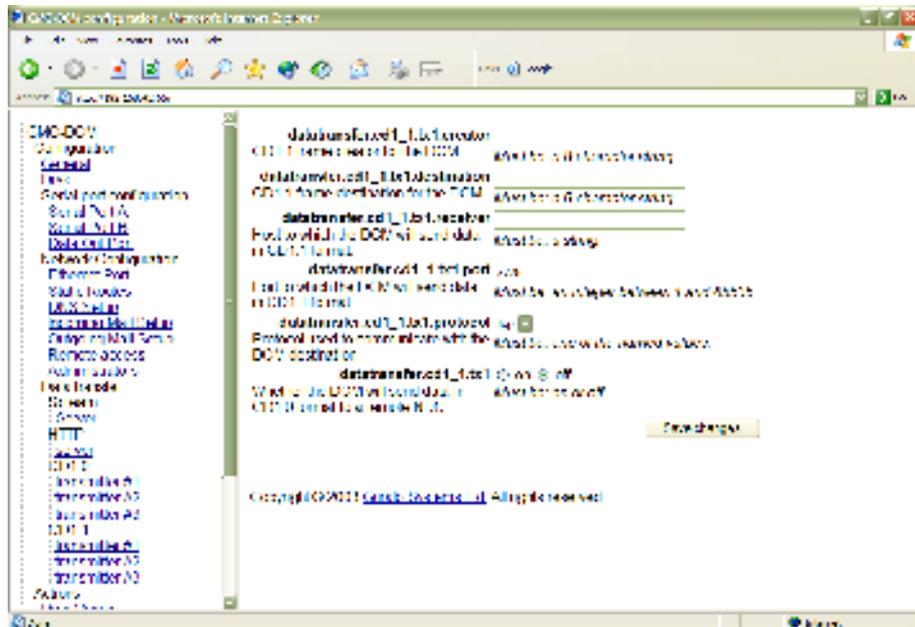
datatransfer.cd1.sender.receiver : The IP address or hostname of the remote CD1.0 device.

datatransfer.cd1.sender.port : The network port number to which the DCM will send CD1.0 data.

datatransfer.cd1.sender.compression : Select *on* to enable Canadian compression in the outgoing data stream.

datatransfer.cd1.sender.channel.x : The DCM needs to know the CD1.0 channel name for each of the streams you want to transmit. When data comes in on a stream that the DCM recognizes, it will send it on with this new name. You can set up to 12 channels to be transmitted by one CD1.0 sender.

CD1.1



The CD1.1 format is an evolution of CD1.0 used by many NDCs, using different settings from CD1.0. The DCM has a separate transmitter for this format.

datatransfer.cd1_1.tx : Select *on* to instruct the DCM to provide data in this format. If you do not want the DCM to transmit CD1.1 data, select *off* and

ignore the remaining settings.

datatransfer.cd1_1.creator : An 8-character string specifying the source of the data (*i.e.* the instrument or array to which the DCM is attached.)

datatransfer.cd1_1.destination : An 8-character string specifying the intended destination of the data.

datatransfer.cd1_1.port : The network port number to which the DCM will send CD1.1 data.

datatransfer.cd1_1.protocol : The network protocol the DCM should use to send CD1.1 data—currently only *tcp* is supported.

datatransfer.cd1_1.receiver : The IP address or hostname of the remote CD1.1 device.

CD1.1 subframe configuration

CD1.1 data is organised into subframes, which have some flexibility in their format. A configuration file, */etc/cd11sf.cfg*, is provided allowing you to specify the exact format to be used in outgoing CD1.1 subframes. Clicking *CD1.1 subframe configuration* brings up a page in the work area 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.

Each line in */etc/cd11sf.cfg* describes a single CD1.1 stream in the format

data-stream:*status-stream*:*location*:*site*:*instrument*:*channel*:*prefix*:
key-bucket

where the fields, separated by colons, are:

data-stream : the digitizer's data stream ID;

status-stream : the digitizer's status stream ID;

location : the CD1.1 location code for the array;

site : the CD1.1 site code for the instrument;

channel : the CD1.1 channel name;

prefix : a prefix for the location to store the files in the file tree of the DCM.

key-bucket : the key-bucket code, which tells the DCM which key to use for that particular stream.

The cryptographic hardware required to produce authenticated CD1.1 streams is installed only in Authentication Modules (AMs). To transmit unauthenticated CD1.1 data, you should use a key-bucket code of -1.

prefix : A prefix 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.

Blank lines, and any lines beginning with #, are ignored.

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*.

DSS

DSS (Data Subscription Service) is a packet format which enables data and statistics to be requested from a seismic installation. A DSS server is designed to handle many concurrent requests from clients with varying levels of privilege, and may prioritize requests according to their origin and urgency.

Güralp Systems' data modules include a package, `libdss`, which is designed to communicate with installations using DSS as either a server or a client. A daemon utility, `dssserver`, is also available which receives requests on a network port and replies to them. This server can be loaded onto Güralp DCM and AM data modules as required.

The `dssserver` program

`dssserver` initially listens on a network port for `DSS_REG` registration packets from clients. Any incoming connections other than `DSS_REG` are refused with a `DSS_REF` packet. Receiving a valid `DSS_REG` packet, `dssserver` sends a `DSS_ACK` packet and adds the client to an internal client list, identified by its source IP address and port number.

Once a client is registered, `dssserver` will accept other DSS command packets from it.

`dssserver` fulfils requests by following this procedure:

- Receiving a valid `DSS_REQ` packet, `dssserver` will send a `DSS_ACK` packet immediately and add the request to an internal

subscription list.

- Receiving a valid DSS_DEL packet, `dssserver` will find the corresponding request in the subscription list, remove it, and reply with a DSS_ACK packet. If the relevant data has already been transmitted, `dssserver` will still send the DSS_ACK packet to notify the client that the request has been dealt with.
- `dssserver` keeps a record of the subscriptions for each client so that it can respond to DSS_LRQ packets. When a subscription is fulfilled, it is removed from the relevant list.
- Other valid DSS commands are dealt with according to the standard.
- When data arrives from connected digitizers, `dssserver` examines the current queue of data requests, and generates the required data or averages separately for each client, in accordance with these requests. This data is placed in a *train*, which is a data structure whose length is determined by the reporting interval. Every train includes data from the vertical, N/S, and E/W component of a particular instrument.
- As soon as the received or generated data fills the buffer, it is ready for dispatch. `dssserver` forms the data into a DSS_DAT message and sends it to the client, identifying it with the relevant *Data Identification Word*.
- Every 10 minutes or thereabouts, `dssserver` checks the client list for any clients which have not reported to it in the past 10 minutes, and purges the inactive clients from the list.

`dssserver` does not implement advanced prioritization or queuing schemes. All clients connect with the same password.

Server configuration

`dssserver` needs to be configured before it can run. On a DCM, this can be done by editing the text file `/etc/dssserver.cfg`, which determines the mapping between incoming GCF streams and DSS stream identifiers. This file can be edited using the DCM's Web page interface, by browsing to the *Configuration* → *Data Transfer* → *DSS* → *dssserver.cfg* page. Each line has the format

stream-id:station-name:location-name:seedname

where *stream-id* is the Guralp stream ID, and the remaining three fields are the stream identifiers used in DSS_REQ packets relating to that stream. A

typical line from the `/etc/dss.cfg` file might thus look like

```
HPAAZ0:VCXX:VC:BHN
```

The station name and location name must be equal to or less than 4 and 2 characters long respectively.

The seedname should be a three-character string specified according to the SEED channel naming conventions. In particular, seednames for vertical components should end in Z, those for N/S components in N, and those for E/W components in E.

The DCM's configuration database includes three further options relating to `dssserver`, which can be altered from the *Configuration* → *Data Transfer* → *DSS* → *Server* page.

datatransfer.dss.server : Set this to *on* to run the server; *off* to disable it.

datatransfer.dss.server.port : The port number on which to listen for incoming DSS packets.

datatransfer.dss.server.password : The password all clients should use when registering with `dssserver`.

datatransfer.dss.server.options : Any additional options to send to the DSS server process (for advanced usage.)

Remember to click *Save changes* before you browse away from the page, if you want to keep the changes you have made.

Restarting the DSS server

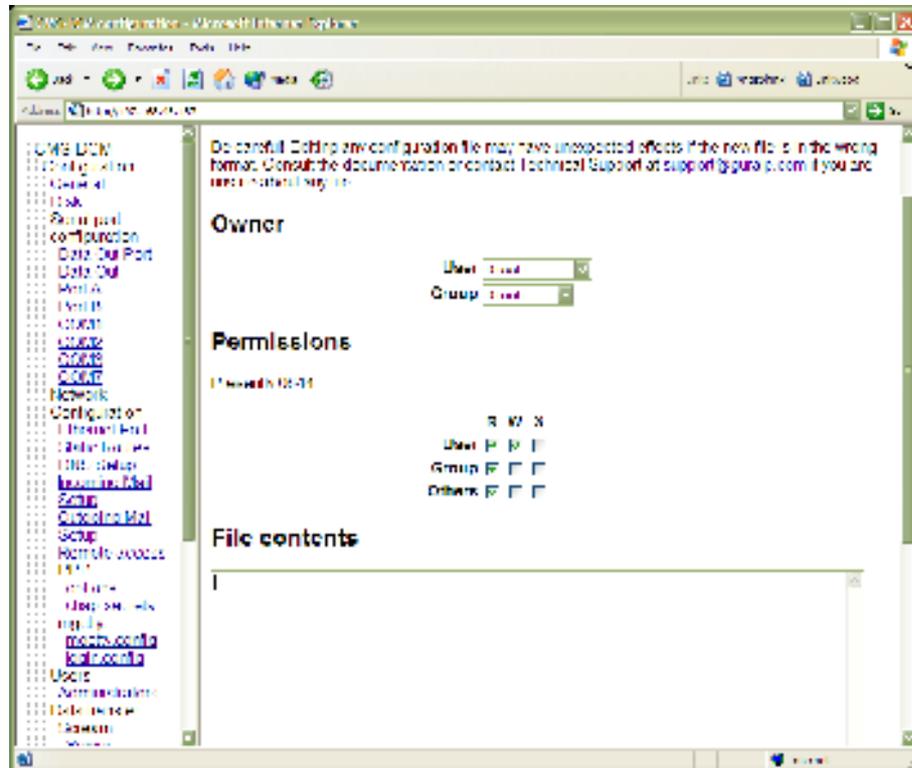
To force the DSS server to restart, select *Configuration* → *Data Transfer* → *DSS* → *Restart DSS Server*, and click the *Yes* button. This action will immediately kill any running DSS servers, and currently-activated subscriptions will be terminated without notice. Clients will have to register with the new instance of `dssserver` before they can continue receiving data.

SEED

The DCM can compile full SEED volumes in real time. To enable this, select *yes* for the *disk.recordas.seed* option under **Configuration** → **Disk** (see above.)

A configuration file, `/etc/seed.cfg`, is provided allowing you to provide technical information about the streams to be recorded in SEED format. **The DCM will only write SEED volumes for streams mentioned in this file.** Clicking **SEED configuration** brings up a page in the work area 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:::::CMG-3_120S_50HZ:CMG-DM24mk3:V9.800E-01:1.6E-07:46
HPA1N2:::::CMG-3_120S_50HZ:CMG-DM24mk3:V9.400E-1:1.6E-07:46
HPA1E2:::::CMG-3_120S_50HZ:CMG-DM24mk3:V1.050E00:1.6E-07:46
```

Each line represents an incoming stream from the digitizer; the fields are separated by colons, and are in turn:

- the digitizer's stream ID;
- a SEED network name for the array (or leave blank to have the DCM generate one automatically);
- a SEED station name for the station (or leave blank to have the DCM generate one automatically);
- a SEED location name for the instrument (or leave blank to have the DCM generate one automatically);
- a SEED channel name for the stream (or leave blank to have the DCM generate one automatically);

- a case-sensitive code for the sensor type (see [Appendix B.2, “Digitizer type codes”](#));
- a case-sensitive code for the digitizer type (see [Appendix B.2, “Digitizer type codes”](#));
- the sensitivity of the sensor, prefixed with a V for a sensitivity in V/m/s, or A for a sensitivity in V/m/s² (in the example, around 1 V/m/s). Note that there is *no* colon between the sensitivity and its unit.

The sensitivity is given on the sensor's calibration sheet. The sensitivity of a sensor integrated with a digitizer is quoted as a *single-ended* sensitivity, whilst stand-alone digitizers are provided with their *differential* sensitivity. A differential sensitivity is quoted as, *e.g.* 2×3000 V/m/s. In this file, you should use the doubled value, *i.e.* 6000 V/m/s, for a stand-alone digitizer. For integrated digitizers, you should use the single-ended value quoted.

- the sensitivity of the digitizer, in volts per count (here 1.6×10^{-7} V/count = 0.16 μ V/count). This is given on the digitizer's calibration sheet.
- the decimation sequence code. This is an integer between 0 and 239 which describes the sequence of filters in use by the digitizer. If you have a recent digitizer (with firmware newer than version 0.91) this information is included in the incoming GCF streams, so you can leave this field blank. The codes corresponding to each possible decimation sequence are available from the support section of the Guralp Systems Web-site.

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 SEED service is restarted. You can restart the SEED volume generator by clicking **Restart SEED recorder**.

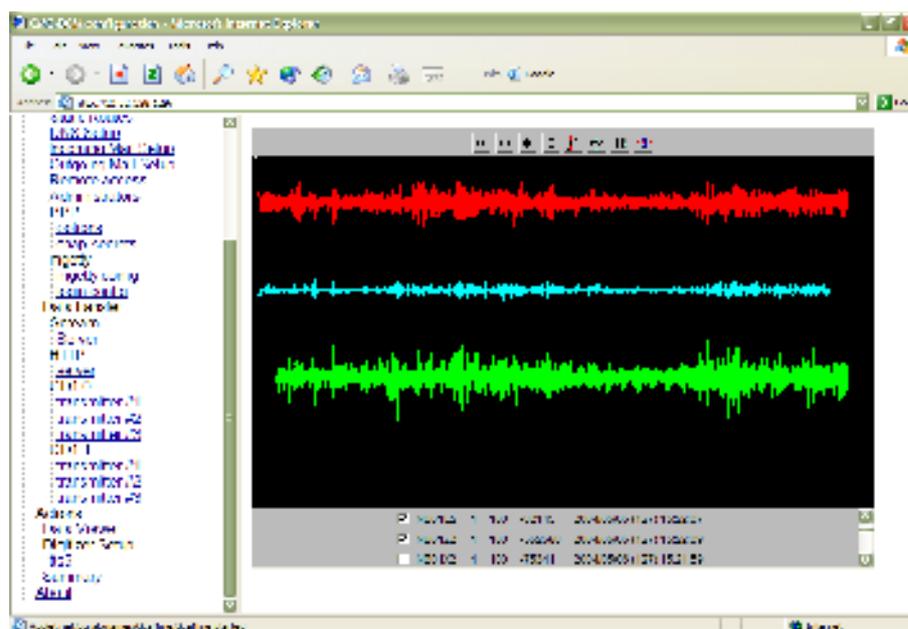
5 Actions

5.1 Data Viewer

The *Data viewer* page uses a simple Java applet to enable you to check that the DCM is receiving data correctly. Your browser must have Java installed for you to be able to do this.

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.

The main part of the display shows a selection of the data streams being recorded, whilst the bars at the top and bottom allow you to control what is displayed, and in what manner:



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 icon depicting a “0” and a sine wave in the icon bar. Otherwise, you can zoom out until the streams are visible.

The icon bar

The bar at the top of the applet allows you to alter how the data is displayed. If you are familiar with Güralp Systems' Scream! package, you will recognise the icons in use here. There are:

- four icons allowing you to alter the X and Y scaling factors of the graph (from left to right: magnify X, reduce X, magnify Y, and reduce Y)
- a bandpass filter toggle (not currently available),
- a toggle to display the endpoints of each GCF block as a white line,
- a pause button, which will cause the viewer to stop scrolling as it receives new data, and
- a zeroing button, which attempts to centre all the selected streams in the display.

The stream list

Below the display is a list of all the streams currently being recorded by the DCM. Each has a checkbox indicating whether that stream is currently being displayed.

To the right of the checkboxes, you are given information about each stream, similar to that found in the main window of Scream!. The information provided is:

- The ID of the stream (six characters long);
- The compression code corresponding to the bit depth of the data. 0 = 8-bit data, 1 = 16-bit data, 2 = 24-bit data, and 3 = 32-bit data;
- A recent FIC for the data. The FIC gives a rough idea of the magnitude of the signal being received.
- Finally, the time the most recent data block was received, including, in brackets, the day number within the year (1 = January 1, etc.)

alter the settings on the digitizer to reflect your choices; this done, you should see the message New configuration successfully saved to attached instrument.

If the DCM is connected to a PC running Güralp Systems' Scream! software, you can also configure the digitizers from within Scream!. See the User Guide for your digitizer model for more details.

General digitizer settings

Baud rate : The speed at which the digitizer will communicate with the DCM, in bytes per second. This must match the baud rate the DCM is using for the serial port linked to this digitizer. The DCM's baud rates can be altered on the various *Serial port configuration* pages: see [Section 4.3, “Serial port configuration”](#).

You should ensure that the baud rate is high enough to allow all the data to be transmitted at the rates you have chosen. As an example, for three streams transmitting at 100 Hz, a rate of 9600 baud is usually sufficient. Modern modems can normally operate at rates up to 57600 baud (~56 kbits/s), although the telephone or transmission lines may not support such a high rate. The same is true of radio telemetry links.

System ID and serial number : Together, these two fields uniquely identify data originating from a particular instrument. Every data or status block sent by the digitizer will contain them as the first two 32-bit fields in the header, each encoded as a sequence of 6 base-36 numbers representing alphanumeric characters. On delivery of the digitizer from the factory, the system ID is set to the Güralp Systems works number, and the serial number is unique for that digitizer (unless shared with an attached seismometer). You may reset these values to any combination of up to 6 letters and numbers. For example, you may wish to set the system ID to a more easily-recognised value, such as an abbreviation of your institution's name.

Sensor type : If the sensor attached to the digitizer is a Güralp velocity sensor, useful seismometer functions (such as sensor locking, centering, and calibration) may be controlled through the digitizer and DCM. In order to be able to do this, and to know which functions to make available, the digitizer must be set up with the correct sensor type: one of *CMG-40T*, *CMG-3ESP*, or *CMG-3T*. The sensor's type will be printed on its casing. If you have taken delivery of a DCM already bonded to a sensor and digitizer, this field will have been set for you at the factory, and should not be altered.

Timing source : The digitizer needs to be able to time-stamp accurately all data that passes through it. It can set its clock either by receiving time signals from the GPS satellite network using an attached Garmin GPS unit, or by taking time information from a central site via the DCM (*stream sync* mode). In *stream sync* mode, the digitizer expects to receive GCF packets from the central timing

source (which may have its own GPS unit, or take signals from one of the radio time standards). The DCM can recognise GCF timing packets and will pass them on to all connected digitizers.

GPS Power Cycle : If you have selected *Garmin GPS* as the timing source, above, this setting determines how often the attached instrument will power up the GPS receiver to obtain an accurate timing signal. Between timing fixes the instrument will run on its internal clock, saving power at a small expense in accuracy. If your instrument has ready access to a power source, you should select *always on*.

Digitizer output control

The analogue-to-digital converters on a Güralp digitizer output data sampled at 2000 Hz, which is then filtered and reduced to a lower rate (*decimated*) using an on-board digital signal processing (DSP) unit. The DSP has four filtering-decimation stages, which run one after the other. Each can be programmed to reduce the sampling rate by a factor between 1 and 10. The output of each stage is called a *tap*. Each of the taps may be configured for a different decimation factor by choosing values from the drop-down menus on the left. If you are using a mouse wheel to select values from a drop-down menu, ensure you remove the focus from the drop-down menu before scrolling the window, or you may inadvertently change the setting.

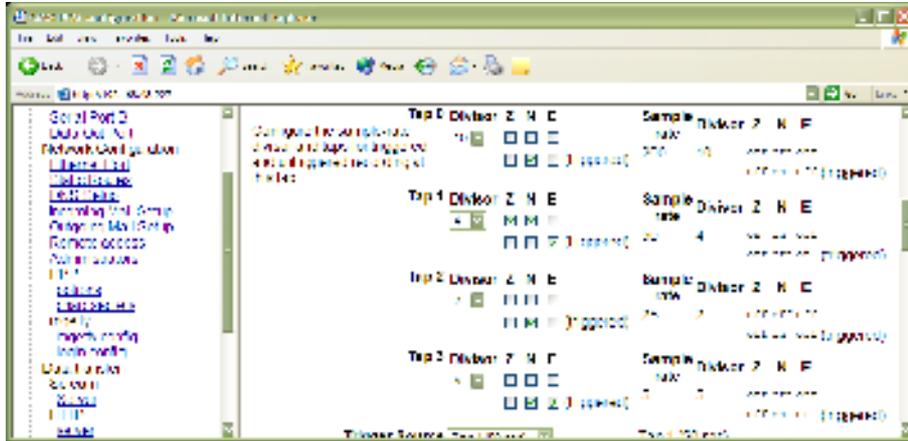
Note that not all digitizers support the full range of taps and decimation factors. For example, the Güralp DM24 allows you to select decimation factors of 2, 4, 5, 8, and 10 only, and does not allow the decimation factor of Tap 0 to be altered from its default setting of 10. In addition, no combination of decimation factors may be used which produces a non-integer data rate (in Hz). A full list of possible tap combinations for the DM24 is given in [“Trigger criteria” in Section 5.2](#), below.

To the right of each decimation factor menu is a grid of six check-boxes marked *Z*, *N*, and *E*. These boxes mark which streams of data to record at each sample rate. Three streams of data are measured by the seismometer, corresponding to movement along each of three perpendicular axes. Although all the streams are decimated by the same set of successive scale factors, you can decide at which stage(s) of processing each stream outputs data. A tick in one of the check boxes will produce an output for a particular channel (column) at the corresponding sample rate (grid).

Each grid also has two rows, which differentiate between constant and triggered output. If a box in the upper row is ticked, that stream will produce output constantly at the corresponding sample rate. If the box below it is ticked, that stream will only produce output at that rate *if* a particular set of trigger criteria are also met. If the constant-output check box is ticked, the other will be ignored.

The table to the right shows the current setting of the digitizer.

For example:



In this example, under normal conditions,

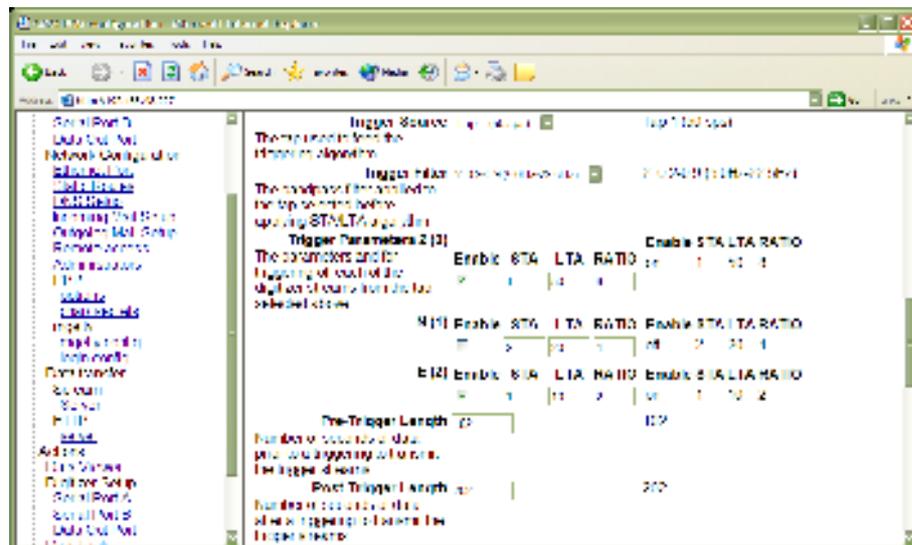
- the Z and N data streams will be output by Tap 1 at a rate of $2000/10/4 = 50$ Hz, and
- the E data stream will not be output.

Additionally, when the trigger criteria are met,

- the N data stream will be output by Taps 0, 2 and 3 at rates of $2000/10 = 200$ Hz, $2000/10/4/2 = 25$ Hz and $2000/10/4/2/5 = 5$ Hz, and
- the E data stream will be output by Taps 1 and 3 at rates of $2000/10/4 = 50$ Hz and $2000/10/4/2/5 = 5$ Hz.

The next section allows you to alter the criteria that the digitizer uses when deciding whether a trigger event has occurred.

Trigger criteria



The triggering algorithm applies a simple short-term average / long-term average calculation to the triggering stream. It works by identifying sections of an incoming data stream when the signal amplitude increases. The purpose of taking a short term average, rather than triggering on signal amplitude directly, is to make it less likely that spurious spikes will trigger the device. Averaging also introduces an element of frequency selectivity into the triggering process.

You can select which tap is tested for the trigger from the **Trigger source** drop-down menu. The tap does *not* have to be selected for data output for you to be able to use it here.

The next option, **Trigger filter**, allows you to apply a bandpass filter at this stage (see below.)

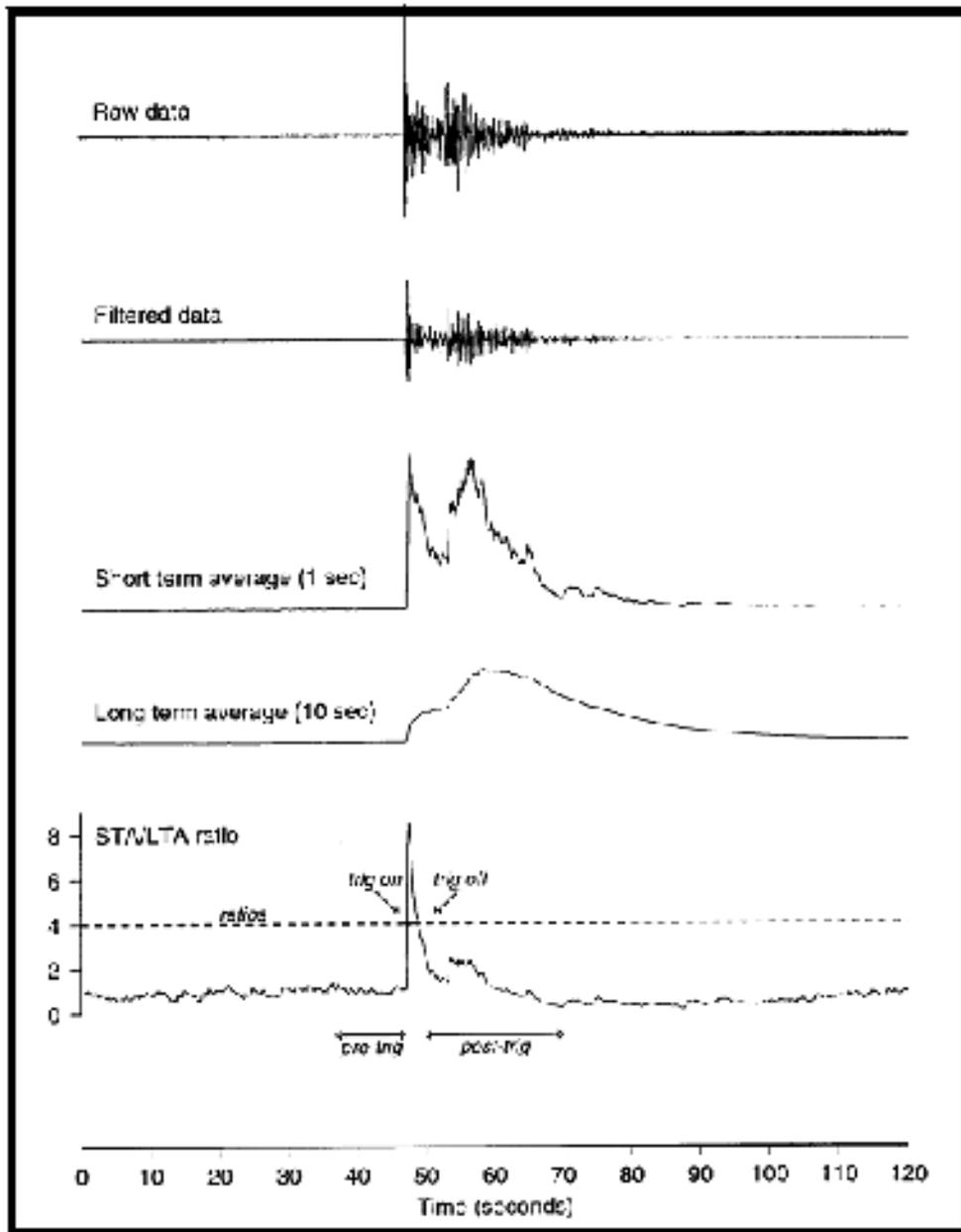
Any or all of the channels available at the tap you have selected may be used to determine a trigger. The next part of the window lists the channels, each with **Enable**, **STA**, **LTA** and **Ratio** settings. The **Enable** boxes determine which channels are considered for triggering. If *any* of the checked channels passes the trigger condition, the trigger will activate, and will not dettrigger until *all* of the checked channels have fallen below their respective **Ratio** values.

The **STA** and **LTA** columns allow you to set the intervals over which the two averages are calculated, in seconds. Typically, the time interval for the short term average should be about as long as the signals you want to trigger on, while the long term average should be taken over a much longer interval. Both the STA and LTA values are recalculated continually, even during a trigger.

The **Ratio** column determines by what factor the STA and LTA must differ for the trigger to be passed. Finding the ratio most suited to your needs is best done by experiment. Too high a value will result in events being missed, while too

low a value will result in spurious non-seismic noise triggering the system. Like the averages, their ratio is continuously recalculated for all components. Note that none of the boxes are allowed to be empty, and so you will need to enter the new value before removing the old one. Alternatively, you can use the up and down cursor keys to change the values.

For example, setting the **STA** to 1 second, the **LTA** to 10 seconds and the **Ratio** to 4 would give rise to the following trigger behaviour:



Usually, the values of the **STA** and **LTA** periods, and of the **Ratio**, will be the same for all checked channels. For convenience, **Scream!** will automatically fill in other values to match ones you enter. If you want to use different values for

some channels, you should uncheck **Common values** before altering them.

If you are using Scream!, you can use the *Control* window to change the values of the **STA** and **LTA** periods, together with the **Ratio**, without restarting the digitizer. See the documentation for Scream! for more details.

Since it is not generally advisable to trigger from broadband data, the digitizer provides a set of standard bandpass filters to apply to the data streams before they are tested for the trigger condition. This filtering serves to maximise sensitivity within a the frequency band of interest, and filter out noise outside this band. You can select which bandpass filter to use from the **Trigger filter** drop-down menu. The corner frequencies of the pass band of the filter are determined by the Nyquist frequency, which is given by the sampling rate of the triggering data. The three filter options have pass bands between 10 % and 90 %, between 20 % and 90 % and between 50% and 90% of the data's Nyquist frequency, respectively.

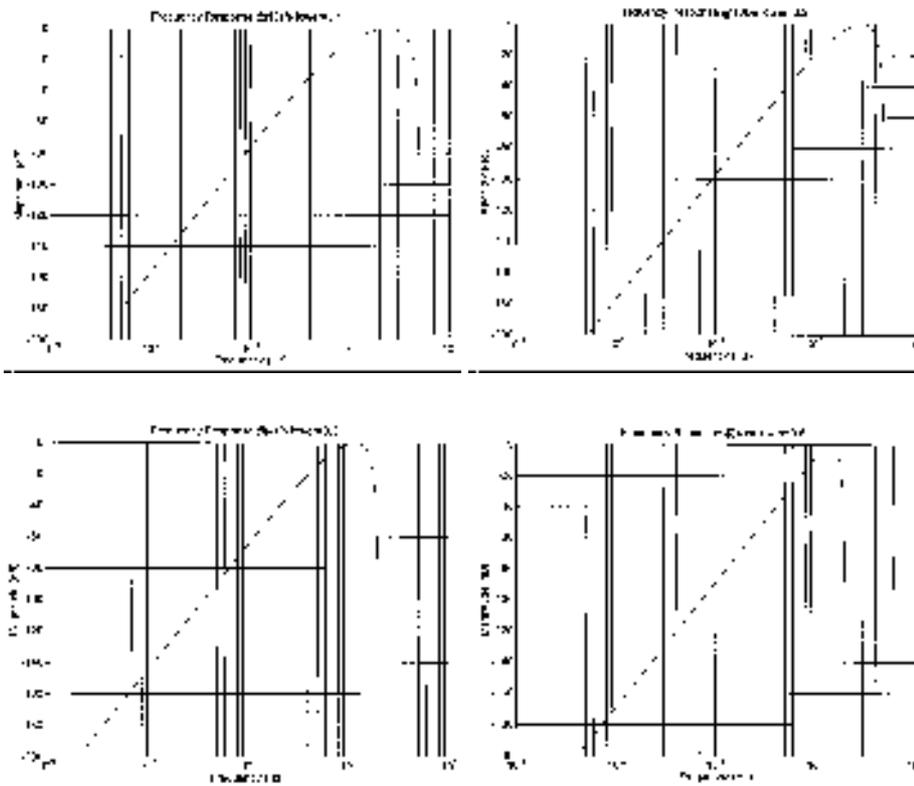
The possible filter configurations are shown in the following table:

Tap #	Rate (samples/s)	Bandwidth 1 (Hz)	Bandwidth 2 (Hz)	Bandwidth 5 (Hz)
0	200	10 – 90	20 – 90	50 – 90
1	100	5 – 45	10 – 45	25 – 45
	50	2.5 – 22.5	5 – 22.5	12.5 – 22.5
	40	2 – 18	4 – 18	10 – 18
	25	1.25 – 11.25	2.5 – 11.25	6.25 – 11.25
	20	1 – 9	2 – 9	5 – 9
	2	50	2.5 – 22.5	5 – 22.5
25		1.25 – 11.25	2.5 – 11.25	6.25 – 11.25
20		1 – 9	2 – 9	5 – 9
10		0.5 – 4.5	1 – 4.5	2.5 – 4.5
8		0.4 – 3.6	0.8 – 3.6	2 – 3.6
5		0.25 – 2.25	0.5 – 2.25	1.25 – 2.25
4		0.2 – 1.8	0.4 – 1.8	1 – 1.8
2		0.1 – 0.9	0.2 – 0.9	0.5 – 0.9
3	25	1.25 – 11.25	12.5 – 11.25	6.25 – 11.25
	10	0.5 – 4.5	1 – 4.5	2.5 – 4.5
	5	0.25 – 2.25	0.5 – 2.25	1.25 – 2.25
	4	0.2 – 1.8	0.4 – 1.8	1 – 1.8

Tap #	Rate (samples/s)	Bandwidth 1 (Hz)	Bandwidth 2 (Hz)	Bandwidth 5 (Hz)
	2	0.1 – 0.9	0.2 – 0.9	0.5 – 0.9
	1	0.05 – 0.45	0.1 – 0.45	0.25 – 0.45

As can be seen, the filter you choose defines the set of permissible sample rates.

The spectral amplitudes for the various frequency responses available are shown in the figures below.



Auxiliary (“Mux”) channels

Güralp digitizers provide a range of slow-rate auxiliary channels for reporting the system's state of health and other diagnostic information, known as multiplexed (“Mux”) channels. The number of Mux channels depends on the model and configuration of your digitizer. Generally, three channels are used to report the sensor mass position, and another measures the internal temperature of the digitizer. In addition to these, up to 12 Mux channels may be supplied for the user's own purposes. Some digitizers have a separate *AUXILIARY* port which can be used to access these channels.

You can choose which, if any, of these channels should be transmitted to the DCM in the next section of the page. If one of the check-boxes is ticked, the

digitizer will output data on that channel to the DCM, which will then store or transmit it with the rest of the data, according to the way it is configured.

Since it is an optional feature, the digitizer may not use the *Pressure Mux* channel to report pressure data. If this is the case, that channel may be used for another purpose. Likewise, the three channels marked *Spare* may not be used, depending on the optional features present in the instrument.

For more information on the format of data packets transmitted on the Mux channels, please refer to the documentation supplied with your digitizer.

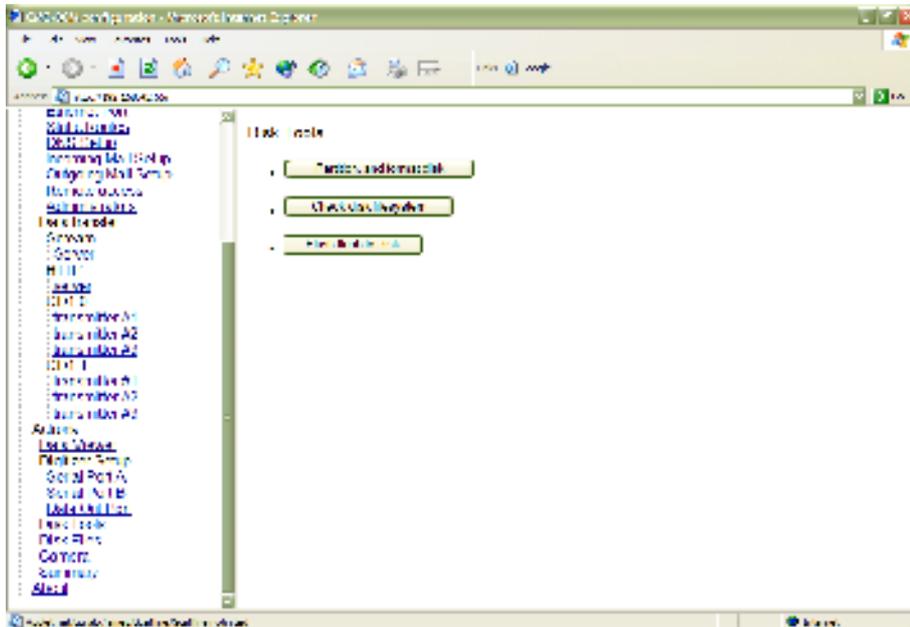
Sensor mass control

There are three commands which can be relayed to an instrument through the DCM to control the position of its sensor mass. For each instrument the buttons which perform these commands (*Centre instrument*, *Lock instrument*, and *Unlock instrument*) can be found at the bottom of the relevant *Digitizer Setup* page. The type of sensor you have installed determines which, if any, of these commands have any effect.

- The *CMG-40T* sensor type has no mass lock or centring capabilities, so all three buttons are inactive.
- The *CMG-3ESP* sensor type has manual mass lock and remote centring, so only the *Centre instrument* button is active.
- The *CMG-3T* sensor type is an automated analogue instrument, with all three commands available.
- The *CMG-3TD* sensor type is a fully digital borehole instrument, with all three commands available.

Lock and *Unlock instrument* are provided so that you can secure the seismometer's sensor mass for safe transportation. Once installed and unlocked, you should *Centre instrument* to move the mass to the correct position to start measuring data. During the execution of any of these commands, the system will display the three components of the mass's current position and update them once per second per component. When the masses are locked, the position readings will be in the region of 32,000 or -32,000 counts. When the masses are correctly centered all three readings should be less than $\pm 1,000$ counts. Locking or unlocking the sensor mass typically takes several minutes to complete.

5.3 Disk tools



The buttons on this page allow you to perform some important actions on the DCM's primary hard disk. By default, the module looks for a connected USB hard disk to use as its primary storage medium. If no suitable storage medium can be found, you will see the message `Failed to find a USB disk` when you attempt to perform any of these actions.

Partition, and format disk

Before using the CMG-DCM you should ensure that its primary hard disk is ready to receive data by partitioning and formatting it. The DCM uses a special journalling format for the hard disk which is designed to maintain the integrity of your data at all times. The hard disk is guaranteed to contain a FAT32-compatible filesystem, even if a write operation fails or is aborted suddenly.

Check disk filesystem

Clicking on this button will verify the integrity of the hard disk's filesystem. It is recommended that you do this immediately after installing the device.

Flush flash

Clicking on this button will cause the DCM to dump the current contents of its Flash memory to the hard disk, thus synchronizing it with the most current data. If you want to remove or replace the DCM's hard disk, using this tool will ensure that the outgoing disk is up-to-date. *Flush flash* does *not* remove data from the DCM's on-board Flash memory. If you *Flush flash* and then swap hard disks, the data remaining in memory will later be written out to the new hard disk, causing some overlap between it and the old disk.

Whilst the DCM is copying the contents of the Flash memory to disk, you will be shown a log of its progress. The USB interface allows data transfer at a speed of around 100 Kb/s, so large files may take several minutes to complete. If an error occurs at any point, it will be marked in red.

5.4 Disk files

This link allows you to browse through any files currently on the DCM's primary hard disk.

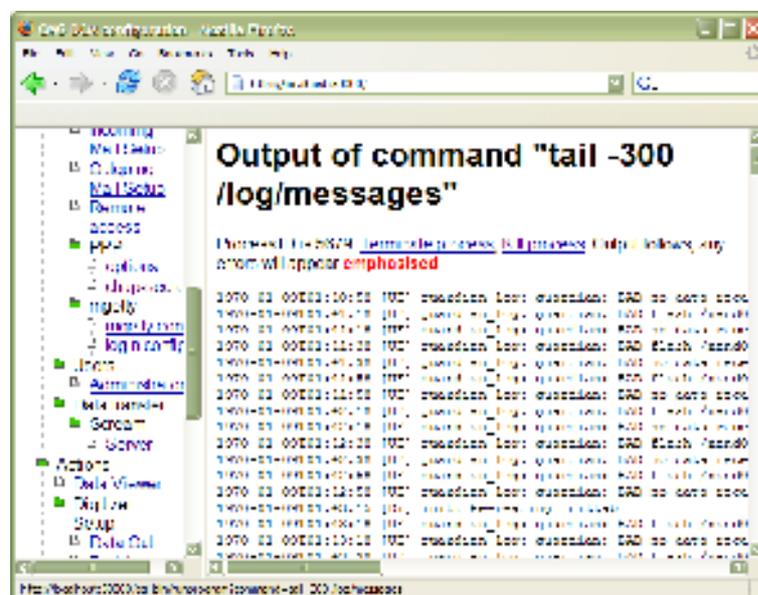
If no suitable storage medium can be found, the module will report the error Failed to open USB disk.

5.5 Camera

If you have attached a compatible camera to the external USB port of the DCM, clicking on this link will show the current view from the installation site. A new image is retrieved every 12 seconds, or whenever you reload the page. Güralp Systems can supply a compatible camera with the DCM, although any STV0680-based device can be used.

5.6 Recent Log Entries

This link displays the most recent 300 entries written to the DCM's internal log file.

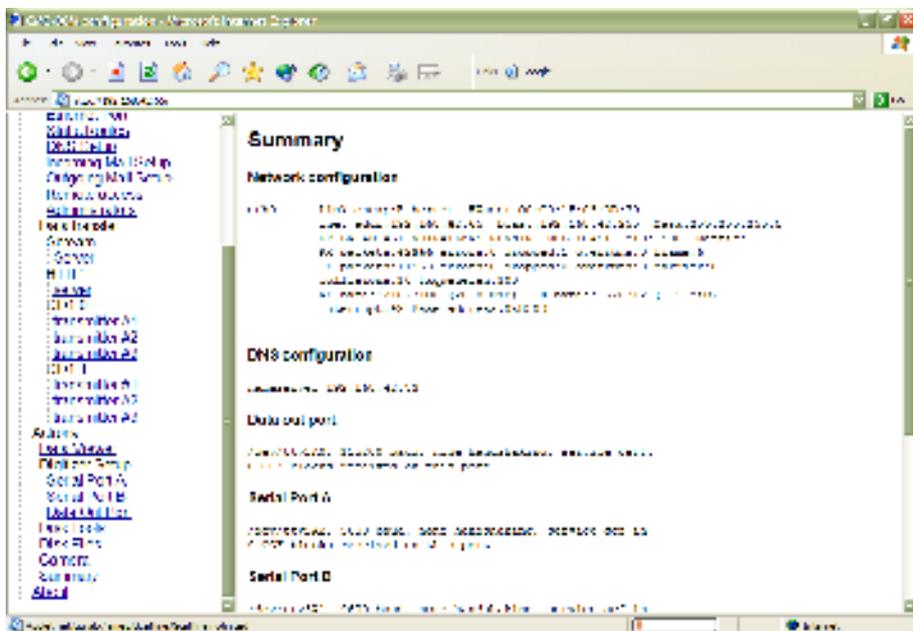


Each entry includes, in order:

- the date and time of the log message,

- a code [Xx] describing the type of program and the importance of the message (e.g. all errors have E as the second letter: see the Linux man page for syslog for full details),
- the name of the program (e.g. guardian_log) which generated the message,
- the text of the message.

5.7 Summary



This link displays a page summarizing the current setup of the DCM. When you first log in to the module over its Web interface, this is the page you are initially presented with. It is divided into a number of sections:

Network configuration

This section displays basic information about the DCM's network setup. It is identical to the output from the Linux `ifconfig` program. A typical reading might look like this:

```
eth0 Link encap:Ethernet HWaddr 00:D0:1F:34:EB:08
inet addr:192.168.0.46 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:3619 errors:0 dropped:0 overruns:0 frame:0
TX packets:2453 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
```

```
RX bytes:290005 (283.2 kb) TX bytes:417231 (407.4 kb)
Interrupt:42 Base address:0x8300
```

For further information, see the Linux manual page for `ifconfig`.

DNS configuration

This section reports the current status of the DCM's domain name resolution service. This is done by presenting the contents of the standard Linux `/etc/resolv.conf` file.

```
# eth0 begin
domain guralp.local
nameserver 192.168.0.1
nameserver 192.168.0.2
# eth0 end
```

For further information, see the Linux manual page for `resolv.conf`.

Data Out Port, Serial Port A, Serial Port B

The next three sections report information about the DCM's serial interfaces. A typical reading for one port might be:

```
/dev/ttySA0, 115200 baud, none handshaking, service getty
0 GCF blocks received on this port
```

In this example, `/dev/ttySA0` represents the device name of this serial port in the Linux filesystem. The remaining information on this line reflects the current setup of the port, based on your choices on the *Serial port configuration* page. See [Section 4.3, “Serial port configuration”](#) for more details.

`115200 baud` : The speed of data exchange over this link, as set up by the *serial.x.baudrate* option.

`none handshaking` : The flow-control protocol used over this link, as set up by the *serial.x.handshaking* option.

`service getty` : The service provided over this link, as set up by the *serial.x.service* option.

The second line reports how many valid blocks of GCF (Güralp Compressed Format) data have been received over this link since the last reset or change in data port configuration. If the link is being used for a service other than GCF data transfer, this number may be disregarded.

Flash Status

This section reports the current contents of the DCM's Flash storage. There may be one or several partitions of Flash memory in use. If a partition is not in use, it will be empty aside from the standard directory `lost+found`. Otherwise, one or more GCF-format files will be shown, together with each file's size in bytes and the time it was created (Unix *ctime*).

Tamper lines

This section, if present, reports the status of the tamper lines relayed to the DCM over an external State of Health interface:

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 DCM was last booted.

Disk Status

This section reports the current state of the DCM's primary hard disk. The module looks on its USB interfaces for an on-board USB hard disk to use as its primary storage medium. If no suitable storage medium can be found, you will see the message `Failed to find a USB disk`.

Assuming that the DCM successfully found a primary hard disk, you will see information in a format like this:

```
FAT32 filesystem has 15 G bytes free
Partition is 37 G bytes (78140097 blocks of 512 bytes)
```

To browse through the disk files, you should use the *Disk files* page, described above.

Software Versions

The final section lists the software currently installed on the DCM's Linux operating system, together with each program's version number. If you need to contact Güralp Systems about any of the installed programs, you should quote the version number in your correspondence.

6 Inside the DCM

The DCM is a fully-functional, Linux-based computer system especially designed for handling seismic data. It can collect and store data from several sources and, if required, output it in your preferred format to other locations on your network or on the Internet. This is done in the following manner.

Firstly, the DCM receives some data from an instrument connected to it. This can be any of

- a digitizer connected through a serial link,
- a computer running a Scream! server,
- a CD1.0 or CD1.1 transmitter (optionally), or
- another DCM or AM.

All the received data is stored in files in the on-board Flash memory. There are two banks of Flash memory available, which are accessible as `/nand0` and `/nand1` in the Linux file tree. Data is normally stored as GCF (Güralp Compressed Format) files.

As an option, you may be able to configure the DCM to use the *miniSEED* or *sac* formats instead (see [Section 4.2, “Disk”](#))

In *automatic* mode, when the Flash memory becomes more than 75% full, the oldest data files are moved to the DCM's primary hard disk until it is less than 50% full. If you prefer, you can configure the DCM to write to the hard disk at set intervals (see [Section 4.2, “Disk”](#).)

Writing to the hard disk is performed robustly, so that no data will be lost if a write is aborted (see [Section 6.1, “File systems”](#).) This means that you can safely swap hardware in and out at any time. Stand-alone DCM modules use off-the-shelf Lacie hard disks, which can be easily removed and installed in most conditions. You can specify other models of IDE / USB or IEEE 1394 2.5” disk at manufacture. If an internal disk is not present, and the module has a USB *host* interface, it will look for hard disks connected to its external USB port.



Once the data is stored on the DCM, whether in Flash memory or on the hard disk, it can be retrieved

- by a remote computer running Güralp Systems' Scream!, or other GCF-compatible software;
- by another DCM or AM, also using GCF;
- by setting up a CD1.0 or CD1.1 transmitter on the DCM;
- by direct file transfer (using SSH, HTTP, HTTPS, etc.,)
- optionally, by requesting the data using SeedLink or AutoDRM.

A PC running Güralp Systems' Scream! software can not only collect data from the DCM, but also configure the module and any instruments attached to it.

You may need to enable and configure some of these methods before you can use them: see [Chapter 4, “Configuration options”](#) for more details.

Most installations of the DCM will not require any more complex setting up than the Web configuration system can offer. However, in some cases you may need to take advantage of the flexibility offered by the underlying Linux operating system.

6.1 File systems

The DCM uses the standard Unix/Linux file naming conventions. The operating system resides in two blocks of Flash memory, mounted on `/` and `/boot`; when the module is powered up, a separate boot loader loads the rest of the operating system.

Once the operating system is loaded, the main Flash memory blocks (where present) are mounted on `/nand0` and `/nand1`. Incoming data, which may be from several sources, is combined into a single stream and placed in one of these blocks (whichever is less full). When in use, you can expect each to be between 50% and 75% full, with several data files present. If the DCM is using GCF as its storage file format (recommended; see [Section 4.2, “Disk”](#)), then each file will be named after the timestamp on its first packet of data, in the following fashion:

`file-yyyymmdd-jjj-s-cccccccc.gcf` where `yyyymmdd` represents the date of the earliest data packet in the file, `jjj` the number of full days elapsed since midnight on January 1, `s` the time segment within the day (each day is divided into eight 3-hour segments), and `cccccccc` a unique hexadecimal code included to ensure filenames do not coincide.

When one of the Flash memory blocks approaches capacity, or after a fixed time period (if you have so configured it; see [Section 4.2, “Disk”](#)) the DCM will automatically move them onto the primary USB hard disk. This may be either an internal Lacie hard disk, or an external drive connected to the module through a USB *client* interface. This disk uses a specially-designed journalled filesystem, which is designed to maintain the integrity of your data at all times. Even if a write operation fails or is aborted suddenly, the disk will still contain a valid filesystem with all previously-saved data intact, which can be read using any driver that supports FAT32.

There is a set of specialised commands which allow you to perform basic tasks on this filesystem:

`gfat32df` : Displays the size of the filesystem, and how much free space remains, in a format similar to this:

```
FAT32 filesystem has 15 G bytes free
Partition is 37 G bytes (78140097 blocks of 512 bytes)
```

If no suitable storage medium can be found, you will see the message `Failed to find a USB disk`.

`diskman` : Ensures that the `/nandx` partitions do not become full by moving files when necessary. This program ordinarily runs constantly in the background. However, a user can use the command `diskman -f` to force the Flash memory to be entirely copied to the USB disk. Typing `diskman -f` is identical to clicking the *Flush flash* button on the *Disk tools* page (see [“Flush flash” in Section 5.3](#)). It does *not* remove data from the Flash memory. If you issue `diskman -f` and then swap hard disks, the data remaining in memory will later be written out to the new hard disk, causing some overlap between it and the old disk.

Whilst the DCM is copying the contents of the Flash memory to disk, you will be shown a log of its progress. The USB interface can transfer data at a speed around 100 Kb/s, so large files may take several minutes to complete.

Once a file has been moved from the Flash memory to disk, any further data received which would otherwise be appended to that file will instead be placed in a new file in the Flash memory. Because of this, a stream may occasionally be fragmented. The *automatic* options are chosen to minimize this likelihood by only moving the oldest files, and by keeping files in Flash memory for a reasonable period of time. If you choose to transfer files to disk more often than this, more files will be fragmented.

`gfat32 ls` : List the files present on the hard disk, with the size of each file.

`gfat32 cpf filename-on-disk destination-filename` : Copy a file from the disk into temporary storage in the Linux filesystem (*e.g.* in your home directory.) Once the file is in the Linux file system, you can modify or convert it using your own scripts running on the DCM, or use programs such as `scp` to transfer it to a remote machine.

`gfat32 cpt source-filename filename-on-disk` : Copy a file from the Linux file system onto the disk.

`gfat32 mv filename new-filename` : Rename a file on the disk.

6.2 Command line tools

The DCM module's Linux operating system can be accessed over a network via SSH. There are many programs available for your computer which implement this protocol: `ssh` is included as part of most Linux and Unix distributions, whilst for Windows `putty` is a reliable free client. `ssh` is essentially a secure version of programs like `rlogin` and `telnet`, and provides a simple command line interface to the device. Access to the DCM by SSH is enabled by default, although you can disable it using the `net.remoteaccess.ssh` configuration option (see [“Remote access” in Section 4.4](#)).

In addition, if you have so configured it, you can connect directly to an RS232 port running the `getty` service.

Once the connection has been set up and you have logged in, you will then be shown a command prompt:

```
~ #
```

By default, your account uses the standard Bourne shell, `sh`. If you prefer, the more advanced shells `ash` and `bash` are also available. Many standard Unix programs are also present: `ls`, `cat`, `more`, `sed`, etc.

The following sections describe how to operate a DCM from the Linux command line, including descriptions of all commands unique to the DCM. Any of these commands can be included in your own shell script files, which can be run as services on the DCM or remotely using a `ssh` connection as required. If you need to compile your own C or FORTRAN programs to be run on the DCM, please contact Gralp Systems for assistance.

6.3 Configuration

Configuring the DCM is automated by a suite of command line tools. These maintain a configuration database and check that all the relevant Linux files are kept up to date. If you alter the standard Linux configuration files, you should bear in mind that these tools will overwrite them without checking that they match the information in the database. Because of this you should use the tools wherever possible rather than editing the files directly. The DCM's Web-based configuration system is just a front-end to these tools.

`gcfgdbls prefix` : Enter this command to find out which configuration options begin with the prefix prefix (case sensitive—all the configuration options are in lower case). The options are listed in alphabetical order. For example:

```
~ # gcfgdbls serial.0
serial.0.baudrate
serial.0.handshaking
serial.0.ppp
serial.0.service
```

`gcfgdbset option-name new-value` : Enter this command to set the value of the named option to new-value. The database will perform a simple type check on your value (for example, to check that certain options are numbers), but will not otherwise make sure that your change makes sense.

`gcfgdbget option-name-or-prefix` : Enter this command to find the current value of the named option. Instead of a single option name, you can also use a prefix (as described above) to find out the values of a range of options. The remaining tools also allow you to use prefixes in place of full option names.

Each option in the database can be marked either as “clean” or as “dirty”. This flag tells the DCM whether the database is currently in sync with the state of the device. Whenever you alter the value of an option using the `gcfgdbset` tool, or using the Web interface, the option is marked as “dirty”; the DCM then alters its configuration, and marks the option as “clean” again, to signify that the change completed successfully. Three more commands are provided to allow you to access this flag:

`gcfgdbmark clean option-name-or-prefix` : Enter this command to mark the named configuration option as “clean” in the database.

`gcfgdbmark dirty option-name-or-prefix` : Enter this command to mark the named configuration option as “dirty” in the database.

`gcfgdbisdirty option-name-or-prefix (option-names-or-prefixes...)` : This command finds out whether any of the named options is marked as “dirty”. If none are marked, the command exits successfully; otherwise, it exits with a failure code. Thus you can use `gcfgdbisdirty` in your own shell scripts, to perform actions depending on the status of the DCM's configuration options.

Note that marking an option as “dirty” will not necessarily lead to any action being taken. For example, you cannot force a service to be restarted using these commands.

Each of the four tools can also take two command-line options:

`gcfgdbxxx -h` : Displays a short usage reminder about the command `gcfgdbxxx`. All other arguments will be ignored.

`gcfgdbxxx -c config-database other-arguments` : Normally, all the `gcfgdbxxx` commands work on the system configuration database. However, you can supply the file name of an alternative database using the `-c` option, in which case the command will be performed on that database instead.

For a detailed description of the configuration options in the database, see [Chapter 4, “Configuration options”](#).

Digitizer console access

The command `gcli` allows you to pass commands directly to the digitizer's console. The syntax is:

```
gcli port-number [-f] [-r] command
```

where port-number refers to the serial port connected to the digitizer. You can use `serialmap` to discover which serial port has which number, as described below. command is a DM24 FORTH command, which may contain several words. For information on the commands available on the DM24, please see its own documentation.

`gcli` will wait for the digitizer command to finish before exiting, and will only output any response from the digitizer afterwards. If you need to issue a command which monitors a value, you will need to connect to the digitizer's console directly using `Scream!` or `minicom`.

If you want to pass special characters to `gcli`, using the `-f` option allows you to use backslash sequences as used by the C command `printf` (e.g. `\t = tab`

character, \n = newline, \r = carriage return, \a = bell, etc.) You can use \r to issue several commands to the digitizer in one session, since the carriage return will cause the digitizer to act on the previous command.

Passing the `-r` option causes the digitizer to reboot automatically once the command is completed. This is useful if you are using `gcli` to change the configuration of the digitizer, since many options require a restart for any changes to take effect.

If you need an interactive session with the digitizer, you can use the Linux terminal program `minicom`, which has been configured specially to cooperate with the DCM's various serial services. You can open a session with a digitizer by issuing the command

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

If the port you specify is set to `gcf_in`, the DCM will automatically interrupt the data flow from the digitizer to allow you to enter commands.

When you have finished your session, press CTRL-a then q. `minicom` will ask you if you want to quit without resetting the connection. Choose `yes` to return the digitizer to data mode.

6.4 Monitoring

Data flow

You can check that the DCM is receiving data either by monitoring the *Summary* page of the on-board Web interface (see [Section 5.7, “Summary”](#)), or from a command prompt using the command `gnblocks`:

```
Key 0x007000: Blocks      0 (Port  0, name Data out port,  
device /dev/ttySA0, baud 115200)  
Key 0x007001: Blocks      0 (Port  1, name Port B, device  
/dev/ttySA1, baud 9600)  
Key 0x007002: Blocks    149 (Port  2, name Port A, device  
/dev/ttySA2, baud 38400)
```

This command shows, for each port:

- the *Key* number (in hexadecimal) of the process on that port which deals with incoming blocks,
- the number of blocks received by that process,
- the internal port number of the port,
- the name you have assigned to it,

- the port's Linux device name, and
- the baud rate currently in operation on the port.

You can query a single port by using the port number or key as arguments to the `gnblocks` command:

```
gnblocks 2
gnblocks 0x7002
gnblocks 28674
```

(In the last example, 28674 is the key ID 0x7002 expressed in decimal: $hex\ 7002 = 7 \times 16^3 + 2 = 28674$.)

Another way to find out the index, key ID, name or device name of a particular serial port is to issue the command `serialmap`. A line will be output for each serial port, in the form

```
Port 0, Key 7000, name tts0, device /dev/ttyS0, baud 115200
```

where `port` and `baud` are in decimal, and `Key` in hexadecimal. The related command `serialmap -k` returns the key ID in decimal.

Digitizer status

The Güralp DM24mk3 digitizer outputs status information as a separate stream. If you have a DM24mk3, you can monitor this stream with the command

```
dm24mk3cads -s port-number
```

Whilst this program is running, it will output any status blocks it receives on that port directly onto your terminal:

```
HPA1CD: Wed Jun 30 14:51:13 2004
gps_fix=2 (0x32)
gps_mode=A (0x41)
gps_control=255 (on)
gps_power=255 (on)
gps_offset=167 ticks
busy_counter=0 ticks
locking=0 unlocking=0 centering=0
calibration V N E
```

The information given, after the stream ID (which will always end in CD), is as follows:

- a time-stamp for the status block;

- whether the GPS has obtained a fix (0 = has not received any data, 1 = has not obtained a fix, 2 = has obtained a 2D fix, and 3 = has obtained a 3D fix);
- the mode the GPS is running in (A = automatic, and M = manual);
- whether or not the system clock is being controlled by the GPS (255 = on);
- whether the GPS is currently powered up (255 = on);
- the current measured offset between GPS and the internal clock, in units of 500 ns;
- the current value of the “busy” counter, which counts down towards zero whilst certain digitizer processes (such as calibration) are active;
- whether the sensor is currently being locked, unlocked, or centred;
- which, if any, of the channels are currently being calibrated (the channel appears with a + if this is the case).

Tamper lines

The command `tamper` provides information about the current state of the DCM tamper lines. Issued with no arguments, it will output the status of all tamper lines once every 10 seconds. Including the `-w` option causes it to exit after printing the status once only:

```
[root@dcm-87AD9C933DE1 ~]# tamper -w
Input  State   Last Closed (Low)           Last Open (High)
  0    open   (never)                    Mon Jan  5 21:50:37 1970
  1    open   (never)                    Mon Jan  5 21:50:37 1970
  2    open   (never)                    Mon Jan  5 21:50:37 1970
  3    open   (never)                    Mon Jan  5 21:50:37 1970
  4    open   (never)                    Mon Jan  5 21:50:37 1970
  5    open   (never)                    Mon Jan  5 21:50:37 1970
  6    open   (never)                    Mon Jan  5 21:50:37 1970
  7    open   (never)                    Mon Jan  5 21:50:37 1970
  8    closed (never)                    (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 output of the `tamper` command is included on the *Summary* page under *Actions* on the DCM Web site.

6.5 Updating the DCM

Over the Internet

The easiest way to ensure that your DCM has all the latest software packages is to update it over the Internet.

The DCM is provided with a simple command-line script, `upgrade`, which checks for new versions of all packages in the distribution on the Guralp Systems Web site and installs them as necessary. It also rebuilds the configuration database to work with the new packages, resets it, and reboots the machine.

Running `upgrade` will restore the DCM to its factory settings, so make sure you will still be able to communicate with it before you issue the command, either

- over a serial link to the *DATA OUT* port (115200 baud, 8 data bits, no parity bit, 1 stop bit, no flow control), or
- over a DHCP-enabled network (using an IP address provided by the DHCP server.)

From the hard disk

If your DCM does not have access to the Internet, you can update its software from a connected USB disk. You will need to make a copy of all the files in the directory <http://www.guralp.net/cmgdcm/feeds>, including all subdirectories, in a directory called `/cmgdcm/feeds` on the hard disk. On Linux, you can do this with the command

```
wget -np -nH -m http://www.guralp.net/cmgdcm/feeds
```

Once the files are on the hard disk, you can install it in a DCM and transfer the packages with the command

```
upgrade gfat32
```

This facility is available with versions 2.10 and greater of the `upgrade` package.

Removing support packages

The DCM as shipped includes a package `gsl-backdoor` which enables Guralp Systems to respond to support issues by remote administration (*e.g.* installing updated firmware or packages.) If you prefer not to allow Guralp Systems access to the DCM you should use the command

```
ipkg remove gsl-backdoor
```

to remove the package. This will not affect the operation of the system in any other way; however, it may prevent Güralp Systems' engineers from being able to assist you in the event of problems.

`ipkg` can also be used for other package management tasks; however, if you remove software from the DCM, or replace packages with versions incompatible with the rest of the system, you risk leaving the unit in an unrecoverable state. We recommend that you use only `upgrade` wherever possible, to ensure that you have a fully tested set of packages.

The firmware

Reinstalling the firmware from scratch is a more involved process, and you should only need to do it if the root or boot partitions of the DCM become corrupted. In this case, you can use the boot loader (which is resident in hardware) to update them. You will need a second DCM or Linux computer attached to the console port in order to do this.

Note that the *firmware* is not the same as the *distribution*. If you want to ensure that your DCM has all the latest software, you should use the `upgrade` command described above. Reinstalling the firmware will reset any changes you have made to the system.

The procedure for installing new firmware depends on which revision of the DCM design you have. All recent DCM units use “MkII” firmware. If you are unsure which hardware type you are using, contact Güralp Systems.

MkII DCM or AM

Mark II DCMs and AMs have a 256 kb bootloader, a 1 Mb kernel image and a 64 Mb file system image. Currently all surface DCM units are the Mark II design.

1. Download the latest revision of the root (and boot, if appropriate) files from the Güralp Systems website at <http://www.guralp.net/software/modules/DCM/>
2. Obtain the `flashdcm` tool from the same site. This is a Linux utility which enables you to access the DCM's firmware. If you are using a second DCM, `flashdcm` will already be present on it.
3. Note the baud rate of the DCM's console port (115200 by default).
4. Power down the DCM.
5. On the second computer or DCM, issue the command

```
flashdcm -r root-file -b boot-file -s baud-rate -p root-  
password -d port
```

where root-file and boot-file are the images you wish to transmit, baud-rate is the baud rate of the DCM's console port, root-password is the password of the user `root` on the DCM (*not* any other administrative account), and port is the port number of the serial interface (on the second computer or DCM) which you will be using.

If the device has several serial interfaces, the command `serialmap` may help you determine which port number corresponds to which device:

```
[root @ dcm] # serialmap  
Library version: libserialmap Version 1.0.5 with  
LIBGCONFIGDB  
4 serial ports  
Port 0, Key 7000, name ttyS0, device /dev/ttyS0, baud  
38400  
Port 1, Key 7001, name ttyS1, device /dev/ttyS1, baud  
19200  
Port 2, Key 7002, name ttyS2, device /dev/ttyS2, baud  
19200  
Port 3, Key 7003, name ttyS3, device /dev/ttyS3, baud  
19200
```

`flashdcm` also has an `-e` option, which wipes clean the Flash memory used for the filesystem *before* providing the new firmware. Some surface DCM units require you to do this when installing a root image. If you use the `-e` option to install a root image, you must also provide a new boot image, since the boot image in memory will be erased.

6. Power up the DCM. The second computer or DCM will detect messages coming from the bootloader of the DCM, and automatically interrupt the boot process to provide the new firmware.
7. Wait for the new firmware to be uploaded. The DCM will reboot automatically at the end of transmission, and the `flashdcm` process will terminate. This may take several minutes.

At this point, you will have a minimal installation of the DCM software, and can proceed to commissioning the system.

Appendix A Connector pinouts

Appendix A.1 Modular DCM units

PORT A and B

This is a standard 10-pin mil-spec socket (02E-12-10S). The pinout is such that the port can be connected to the serial output of a DM24 digitizer using a straight-through cable.

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

DATA OUT port

This is a standard 10-pin mil-spec plug (02E-12-10P). The pinout is the same as the serial output of a DM24 digitizer, allowing you to insert a DCM into a pre-existing installation and maintain connectivity.

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

USB connector

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

Pin	Function
A	+5 V DC (USB Type A pin 1)
B	Data -ve (USB Type A pin 2)
C	Data +ve (USB Type A pin 3)
D	0 V (USB Type A pin 4)
E	Shielding
F	Switched power +10 to +35 V

NETWORK connector

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

Pin	Function
B	Data transmit +ve (RJ45 pin 1)
C	Data receive +ve (RJ45 pin 3)
E	Data receive -ve (RJ45 pin 6)
F	Data transmit -ve (RJ45 pin 2)

Appendix A.2 Integrated DCM units

DM/AM module output

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

Appendix B Sensor and digitizer types

These codes are used in situations where the DCM needs to know the response properties of sensors and digitizers, *e.g.* when compiling full SEED volumes.

If you are unsure about the code you should use, contact Güralp Systems.

Appendix B.1 Sensor response 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

Appendix B.2 Digitizer type codes

Digitizer	Digitizer type code
CMG-DM24 mk2 (3- or 6- channel)	CMG-DM2 4
CMG-3TD or 5TD using DM24 mk2 module	CMG-DM2 4
CMG-DM24S12 (including AMS)	CMG-DM2 4
CMG-DM24 mk3	CMG-DM2 4mk3
CMG-6TD	CMG-6TD
