Siemens Mobility, Traffic Solutions

Sopers Lane, Poole, Dorset, BH17 7ER
United Kingdom
+44 (0)1202 782000
http://www.siemens.co.uk/traffic/en

GVP Reference Manual

PREPARED: Mark Retallack/Jim Ballantine

FUNCTION: Senior Firmware Engineer/Lead Engineer

Part Number	Issue	Change Ref	Date
667/HB/31760/000	1	N/A	Dec 2009
667/HB/31760/000	2	TS005686	Mar 2011

THIS DOCUMENT IS ELECTRONICALLY HELD AND APPROVED

SAFETY WARNINGS

In the interests of health and safety, when using or servicing this equipment, the following instructions must be noted and adhered to:

(i) Only Skilled or Instructed personnel with relevant technical knowledge and experience, who are also familiar with the safety procedures required when dealing with modern electrical or electronic equipment, are to be allowed to use and/or work on the equipment. All work shall be performed in accordance with the Electricity at Work Regulations 1989 and the relevant Highways Agency (DoT) procedures of test and maintenance.

(ii) Such personnel must take heed of all relevant notes, cautions and warnings in this handbook, and any other documents and handbooks associated with the equipment including, but not restricted to, the following:

(a) The equipment must be correctly connected to the specified incoming power supply.

(b) The equipment must be disconnected/isolated from the incoming power supply before removing protective covers or working on any part from which protective covers have been removed.

(c) The equipment contains batteries that must be disposed of in a safe manner. If in doubt of the correct procedure, refer to the Siemens instructions.

CONTENTS

TABLE OF FIGURES

CHANGE HISTORY:

Version	Date	Change	Author
1	Dec-2009	First Issue	Mark Retallack/Jim Ballantine
2	Mar-2011	Update Appendix A to clarify use of GeminiConnect.	Jim Ballantine

1. INTRODUCTION

1.1 PURPOSE

This document is intended as a reference manual for the Gemini^2 Generic Versatile Platform (GVP) web interface and handset commands.

Each application running on GVP can add additional web screens and handset commands, which are described in individual product handbooks. However, this manual describes the web screens and handset commands provided by the basic GVP.

1.2 SCOPE

This document covers the GVP web screens and handset commands. No hardware information or product specific information is included.

1.3 REFERENCE DOCUMENTS

Gemini^2 UTMC VMS and Car Park Handbook ..... .667/HB/31600/000

Gemini^2 UTMC OTU Handbook ..... .667/HB/31601/000

Outstation Support Server (OSS) Handbook..... .667/HB/31760/100

1.4 ABBREVIATIONS AND DEFINITIONS

AC	Alternating Current
ASCII	American Standard Code for Information Interchange
Bit	Binary Digit
BOOTP	Bootstrap Protocol
CCITT	International Co-ordinating Committee for Telephony and Telegraphy
CHAP	Challenge Handshake Authentication Protocol
chat	A scripting language for defining the dialogue between a computer and a modem
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
DC	Direct Current
DCE	Data Communication Equipment
DDNS	Dynamic DNS
DNS	Domain Name System
DSL	Digital Subscriber Line
DST	Daylight Saving Time
Dynamic	Application that can be overwritten by new files
EMC	Electromagnetic Compatibility
eCos	Embedded Configurable Operating System
FLASH	Non-volatile memory that may be programmed under software control
GMT	Greenwich Mean Time
GPRS	General Packet Radio Service

GSM	Global System for Mobile Communication
GV	Generic Versatile
GVP	Generic Versatile Platform
IO	Input and Output
ICG	Inter-Cycle Gap - A Siespace Protocol Parameter
ICMP	Internet Control Message Protocol
IMG	Inter-Message Gap - A Siespace Protocol Parameter
IP	Internet Protocol
JFFS2	Journaling Flash File System Version 2
LAN	Local Area Network
LED	Light Emitting Diode
LMU	Lamp Monitoring Unit
MD5	Message Digest 5
MIB	Management Information Base
MIB-II	Management Information Base II (Two)
NTP	Network Time Protocol
OEM	Other Electrical Manufacturers
OSS	Outstation Support Server
OTU	Outstation Transmission Unit
PAP	Password Authentication Protocol
PC	Personal Computer
PCB	Printed Circuit Board
PIN	Personal Identification Number
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
PSU	Power Supply Unit
RAM	Random Access Memory
ROM	Read Only Memory
ROMFS	ROM File System
RS232	EIA Data Communications Interface - Level based serial communications standard
RS485	EIA Differential Data Communications Interface - Differential serial communications standard
SHA-1	Secure Hash Algorithm 1
Siespace	A Siemens car park guidance system
SNMP	Simple Network Management Protocol
SNMPv1	SNMP Version 1
SNMPv2	SNMP Version 2
SNMPv3	SNMP Version 3
SNTP	Simple Network Time Protocol
STC	Siemens Traffic Controls
Telnet	Telnet Protocol
TfL	Transport for London
TFTP	Trivial File Transfer Protocol
TS	Siemens Traffic Solutions, Mobility Division (formerly Siemens Traffic Controls)
UDP	User Datagram Protocol
UTMC	Urban Traffic Management and Control

UVMS	Urban VMS
VMS	Variable Message Sign
VPN	Virtual Private Network
XML	Extensible Markup Language
ZXO	A Zero Crossover Interface

2. PRODUCT DESCRIPTION

2.1 INTRODUCTION

The UTMC Outstation is a 3U rack mounted unit which is normally located in roadside cabinets. The outstation is used to monitor and control on-street equipment. Several hardware and software versions are available to interface to equipment such as Variable Message Signs, Car Parks and Traffic Controllers

The outstation is controlled from a UTMC instation across an IP network. The IP network employed will be dependent on the system application.

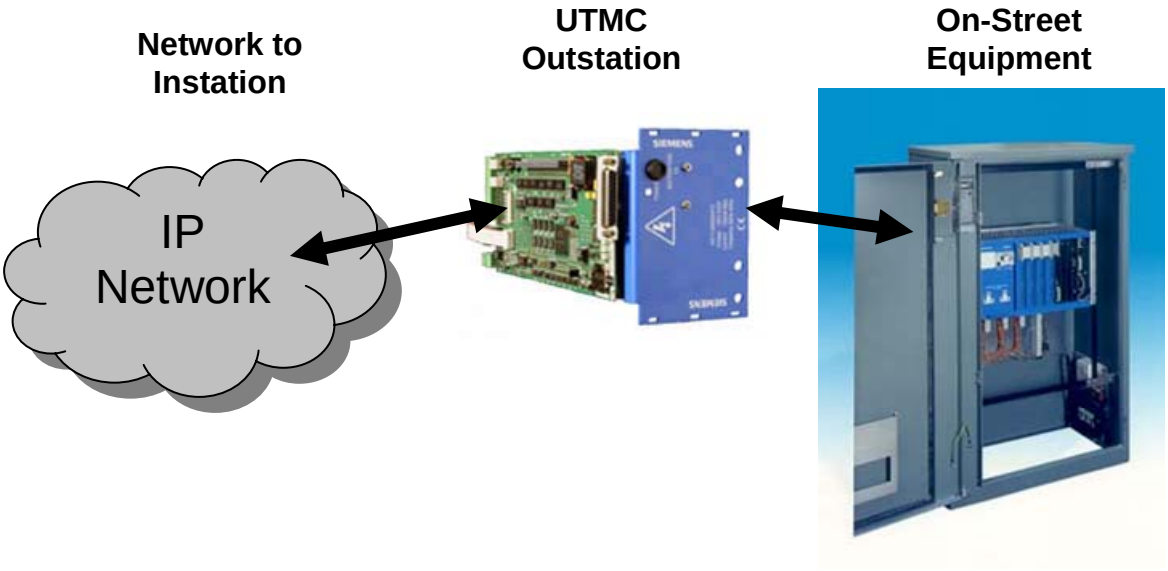


Figure 1 – Typical Outstation

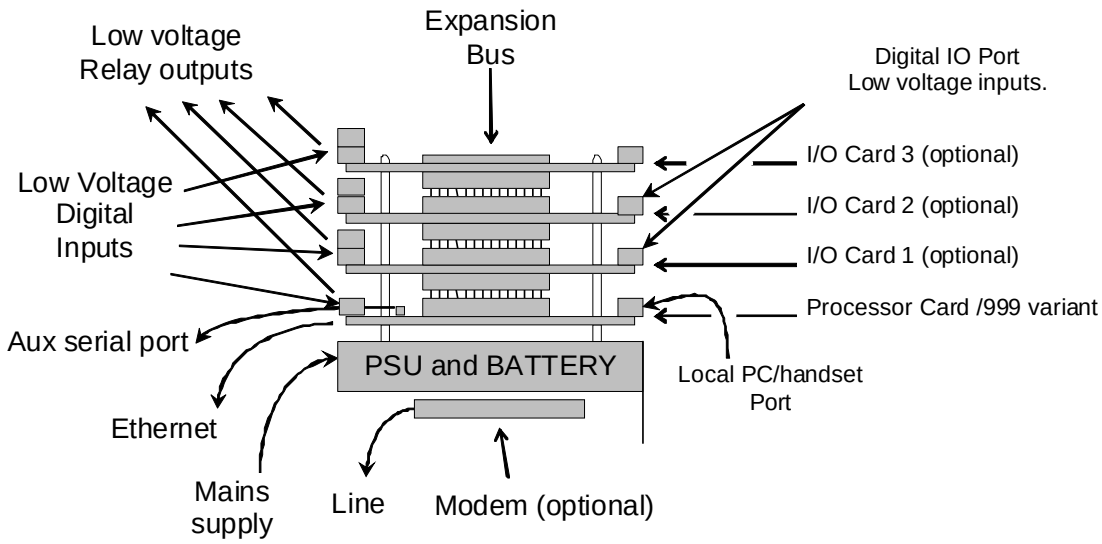


Figure 2 – Hardware Components

The Siemens GVP Outstation has a number of different facilities depending on the Application that is stored in the Flash memory.

The hardware platform is a self-contained unit consisting of a microprocessor based Processor card, up to three IO cards (optional), a PSU, back-up battery and a modem (optional).

The unit is mains powered and is fitted with a battery to support the unit in the event of a mains failure. Assuming the communication equipment is still powered, this allows the outstation to remain in communication with the instation when the mains fails.

2.2 SIEMENS GVP SOFTWARE

2.2.1 The GV Platform

The GVP Outstation is based on the original Gemini^2 Traffic Outstation. The Gemini^2 Processor cards are distributed with the GVP software installed in Flash memory. The flash memory also contains one or more GVP applications (see Figure 3). Different software packages can be loaded, depending on the application required.

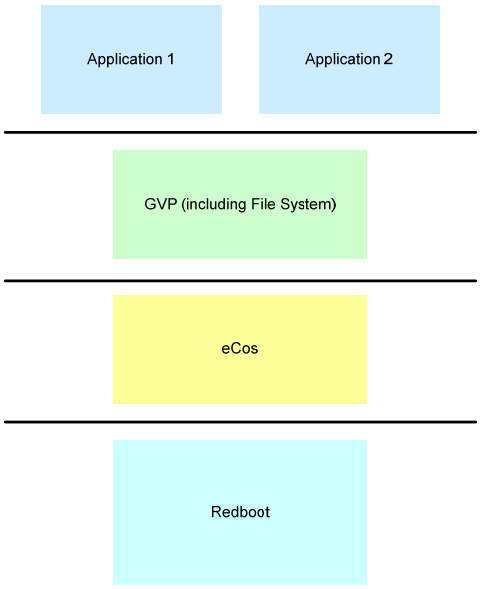


Figure 3 – Software Components

Compatibility
The GVP is compatible with the Generic IO card, the LMU IO card and the BUS/MOVA I/O card.

Software Packages
There are a number of different outstation software packages available, depending on the application. The package part number and issue state can be displayed on the handset using the ‘version` command, see section 3.3.1 for more information.

3. GVP HANDSET COMMANDS

The outstation has a web interface which should be used for most operations – see section 4 for details. The handset command interface can be used to access lower level maintenance and diagnostic facilities.

Commands are not designed for display on hand-held (20 character x 4 line) terminal except for the "WIZ" command and so a 80 character x 25 line terminal emulator such as Hyperterm is recommended.

In general, the GVP commands:

- Have no limits to the length of command names. This means that the commands can be given meaningful names that are easy to remember.
- Are familiar to users of other operating systems, such as `delete`, `pwd` and `ls`.
- Allow parameter values to be defaulted, making them easy to use.

The handset commands are available to terminals or emulators on the Gemini^2 handset port. They are also available via Telnet, if the Telnet service is enabled on the Gemini^2. If a password has been configured then the engineer must enter it before access is given to the handset commands.

When GVP is ready to accept a command it displays a "`> `" prompt. The engineer enters the command and parameters and then presses the `<return>` key. Most commands complete within a few seconds, after which the "`> `" reappears. These commands run to completion; they cannot be interrupted.

Some commands continue until interrupted by the engineer pressing any key. For example, the `din` command continues to display the digital input values until interrupted. See Section 3.8.3.

Command names are not case sensitive. In general, parameter values are case sensitive, so these should be entered precisely. Double quotes around several words ensure that the words are treated as a single parameter. For example, the following commands each have one parameter:

```
helloworld helloworld
helloworld "hello world"
```

The following command has two parameters:

```
helloworld hello world
```

Most commands have a set and show form. Specifying parameters sets new values. Commands specified without parameters display the current configuration values.

In general, the effects of each command can be seen immediately. Where commands affect the operation of the system, they are also stored in a configuration file. After subsequent reboots, the change is still in effect.

This section of the handbook describes all of the GVP handset commands. For each, it gives the command name, its synopsis and details of all accepted parameters. For many of the commands, examples of use are also given. Where commands change the system configuration, the configuration item is also described.

Each GVP application has its own set of commands; these are documented in the handbook for that application.

An important command to remember is `help`. This can be used to list all of the available commands and to display the format and parameters for any of them. As it is so useful, `help` is the first command to be described.

3.1 HELP

The handset commands available on GVP platforms can be displayed using the `help` command. Using `help` without any parameters lists all of the available commands. These are shown in Figure 4. Each command name is displayed along with a brief description of its function.

```
> help


wiz                 Wizard for settings configuration items
oss                 set and display the OSS interface settings
zip                 zip or unzip a file
sitelog             Add, delete site log entries
timeserver          set and display the current time server address
status              get the status of runtime system items
host                perform a dns lookup on a given host
ns                  setup the name server client for dns
ddns                setup the ddns client for dynamic dns
ping                ping the given IP address or host name
ppp                 Show and set the Modem PPP interface settings
ipfw                setup the ip firewall
lwtunnel            setup the light weight tunnel
tftp                put and get file using tftp
httpd               start and stop the http server
hwclock             read the hardware clock
tcl                 run a tcl script
upgrade             start an upgrade
safemode            Set and change safemode settings
loadfile            load a file via handset terminal
genhelp             generate the off-line help docs
dout                sets the current state of a selected outputs
din                 displays the current state of a selected input
modempower          set the modem power state
exit                Force the current CLI interface to exit
reboot              reboot the unit
watch               watch the output from a given command
passwd              set the handset terminal password
configure           setup the configuration of an application
threads             display the current thread list
stop                stop applications
type                display a file
date                set and display the current date and time
cards               display the expansion cards fitted
```

```
version             print the version of the current running firmware
cpuload             display the current load on the cpu in 0.1, 1 and 10
                        second averages
debug               configure and display debug output
start               start applications
help                provides help for available commands
helloworld          print the famous helloworld string
memory              display the current memory usage
cfgdelete           delete a single config item in the database
wipecfg             Wipe the Config
dumpcfg             dump the config file
cfgitem             get or set a single config item in the database
loadcfg             load a new configuration file
diskusage           display the disk usage
mkdir               create a new directory
copy                copy files on the file system
ls                  print the current working directory
delete              delete files on the file system
dirtree             list the complete file system
md5sum              calculate the md5 of a file
sitename            Set and get the site name for this Gemini.
services            set the network services that should be started
sysctl              set and display the sysctl settings
snmpagentport       set and display the current snmp agent IP port number
snmpcommunity       set and display the snmp community string
snmpsetup           set and display the snmp settings
trap                send an snmp trap
snmpuser            set the user and password that snmp should use
snmptrap            set and display the current instation trap ip address and
                        trap type
snmpmibs            display the current registered mibs
route               display the ip routing table
network             configure and display network settings
plugins             list the plugins available
tester              start, stop and find status for tests
>
```

Figure 4 - Using Help to find the available commands

Further details of a specific command can be obtained by specifying the command. As well as describing the commands function, a synopsis of the command's usage is given. See Figure 5.

```
> help debug
debug

usage: debug [tail|log|[set|get all|ModuleName [OutputLevel]]]

configure and display debug output
>
```

Figure 5 - Using Command Help

The usage description shows how the command should be formatted:

1. Parameters shown within square brackets ([,]) are optional.

2. Parameters separated by pipe characters (`|`) are alternatives; only one of the values is allowed.

So, from Figure 5 we can determine that the following commands are all valid:

- `debug tail`
- `debug log`
- `debug get all`
- `debug get DigitalIO`
- `debug set DigitalIO info`

Where `DigitalIO` is a GVP module and `info` is an output level.

The usage format does not distinguish between parameters that must be entered as shown and those parameters that are placeholders for variable data. This section makes this distinction using the following formatting rules:

1. Parameters which have a fixed value are shown in a `mono-spaced` font.
2. Parameters which take a variable value are shown in an *`italic mono-spaced`* font.

So, this section gives the synopsis of the `debug` command as:

`debug [tail|log|[set|get all|`*`ModuleName`* `[`*`OutputLevel`*`]]]`

The following sections also give examples of handset interactions. These examples are also shown in a `mono-spaced` font; user input is shown in a **`bold mono-spaced`** font.

3.2 THE WIZ COMMAND

The WIZ command is designed for use on a handheld 4-line by 20-character terminal. The command starts a “wizard” menu system which can be used to perform the following actions:-

- Display active faults
- Configure the Ethernet
- Upgrade the outstation software from the OSS
- Load outstation configuration file from the OSS
- Display I/O Ports

The wiz command is entered with no parameters:

```
wiz
```

The initial menu is displayed:-

```
1> Active Faults
2> Config/Status
3> Upgrade
4+ Fetch Config
```

Typing **1, 2, 3 or 4** will select that menu option.
Typing <**CR**> will back up one level – at the top level it will exit the wiz command.
Typing **+** will display more menu options, where more than 4 available.
Typing **-** will move back to previous menu options, where more than 4 available.

With the initial menu above, typing **+** will display the additional reboot and digital IO menu options:-

```
1> Active Faults
2> Config/Status
3> Upgrade
4+ Fetch Config
+
1- Reboot
2> Digital IO
```

Each of the commands is described in the following sections.

3.2.1 WIZ: Active Faults

This command will display each entry in the outstation Fault Table (see section 4.2). One fault is displayed at a time. Each time a key is pressed, the next fault in the table is displayed. When there are no more faults to list, the display returns to the initial command menu.

3.2.2 WIZ: Config/Status

This command will display a sub-menu to select either Basic Config or Status:-

```
1> Basic Config
2> Status
```

3.2.3 WIZ: Basic Config

This command uses the menu system to enable the Ethernet configuration parameters to be entered, so that the outstation can be enabled on the IP network. The configuration items are:-

```
1> Eth IP Mode
2> Eth IP Address
3> Ethernet IP Netmask
4+ Ethernet IP Broadcast
+
1- Eth IP Gateway
2> Site Name
3> OSS Address
4+ Enable OSS
+
1- Enable OSS backup
```

3.2.4 WIZ: Status

This command uses the menu system to enable the following status items to be displayed:-

```
1> Free RAM
2> Total RAM
3> Free Disk Blocks
4+ Total Disk Blocks
+
1- CPU load
```

3.2.5 WIZ: Upgrade

This command fetches the latest version of the current outstation firmware package from the OSS. If the outstation is already running the latest version, then no upgrade is performed.

If it is required to switch to a different outstation firmware package, then the web interface upgrade screen should be used, see section 4.3.

3.2.6 WIZ: Fetch Config

This command fetches the latest version of the outstation configuration from the OSS.

3.2.7 WIZ: Reboot

This command reboots the outstation. The software restarts, the fault table is cleared and any applications which were in the ‘Running’ state previously are initialised and re-started. The Site Log, System Log and other application configuration data are maintained over the restart.

3.2.8 WIZ: Digital IO

This command displays the status of the inputs and outputs equipped on the outstation. The names of the available inputs (or outputs) are displayed. When the port type is selected the actual input (or output) line values are displayed in groups of 8. Use the + command to step on to the next group of 8 lines. Use the - command to step on to the previous group of 8 lines.

3.3 SYSTEM COMMANDS

GVP handset commands can be used to examine version numbers of the GVP and the applications. There are commands to reboot the Gemini^2, set the system clock and measure the use of system resources.

3.3.1 Display Product Versions

Details of the Gemini^2 and GVP products are listed by the version command. The synopsis of this command is:

```
Version [-f]
```

The `version` command without any parameters lists the outstation software package part number and version.

The `version` command with the -f parameter lists more detailed information as follows:

`Package Partnumber`	The part number of the outstation software package (which includes gvp and applications)
`Package Version`	The version number of the outstation software package
`Firmware`	The part number and version number of the GVP firmware
`GVP API`	The version of the GVP application programming interface
`Kernel`	The build tag of the GVP firmware
`Bootloader`	The build tag of the bootstrap loader and version of launcher program
`Serial Number`	The serial number of the Gemini^2 Processor card
`Ethernet MAC address`	The Ethernet MAC address of the Gemini^2 Processor card
`Application list`	A list of GVP applications installed on the Gemini^2. This list includes the application name, its part number and version number.

Figure 6 shows example output from the version command.

```
> version
Package Partnumber: 667/TZ/32370/000
Package Version: 1.8
> version -f
Package Partnumber: 667/TZ/32370/000
Package Version: 1.8
Firmware: 667/TZ/31760/000 issue 18.0 (build 1) RELEASE
GVP API: 41
Kernel: REL_GVP_KERNEL_33
Bootloader: REL_REDBOOT_11 + Launcher 4.11
Serial Number: 8183894
Ethernet MAC Address: 0x0:0x30:0xe6:0xfe:0x13:0xa1

Platform Type: Gemini-GVP
Hardware Type: GeminiMK2
Application list:

Tester (000/TZ/00000/000) V0.0.0
Mova1 (667/TZ/32375/000) V1.1.1
Mova2 (667/TZ/32375/000) V1.1.1
UTMCSimpleUTC (667/TZ/32374/000) V1.1.1
UTMCFullUTC (667/TZ/32373/000) V1.2.1
OSEWebConf0 (667/TZ/32376/000) V1.2.3
>
```

Figure 6 - Using `version`

3.3.2 Reboot

The `reboot` command resets the Gemini^2, reloads GVP and starts any previously running applications. The synopsis is:

```
reboot
```

```
> reboot
+

SIEMENS
Password:
```

Figure 7 - Using `reboot`

3.3.3 Display Processor Usage

An indication of the load on the Gemini^2 hardware can be obtained using the `cpuload` command. This calculates the percentage usage of the processor during the previous 0.1 seconds, 1 second and 10 seconds. The synopsis is:

```
cpuload
```

```
> cpuload
0.1 Second average: 0%
1 Second average:   0%
10 Second average:  10%
>
```

Figure 8 - Using `cpuload`

3.3.4 Display Memory Usage

An indication of the load on the Gemini^2 hardware can be obtained using the `memory` command. This shows the amount of used and free memory. The synopsis is:

```
memory
```

The command output shows:

`Memory in use`	The number of bytes of the memory assigned to the heap that is currently in use
`Total heap size`	The number of bytes of Gemini^2 RAM memory assigned to the heap. This is the amount of free memory after the GVP has loaded. This value is fixed after the system has loaded and indicates the memory that can be dynamically allocated.

The heap is used to store application code and data as well as the dynamically allocated data used by GVP.

```
> memory
Memory in use: 677808 bytes
Total heap size: 3468592 bytes
>
```

Figure 9 - Using `memory`

3.3.5 Display Threads

The GVP software is organised into threads. A thread is a processing sequence that runs in parallel with other threads. Typically, each thread supports a particular set of functions. For example the Debug thread is responsible for writing event logs to the log file. Threads may be created by GVP or by GVP applications.

The `threads` command lists the threads that have been defined within the system. The synopsis is:

```
threads
```

This lists all the threads that have been created. For each thread, the following data is shown:

`name`	The name of the thread

used/size The first value is the number of bytes of the thread's stack that are currently in use. The second number is the maximum value of the stack. The stack contains data used by the code running in the thread

priority When several threads are waiting to use the Processor, the thread with the highest *priority* is the first to execute. Priority values are in the range 0 to 31, with 0 being the highest priority.

A `used` value of `-1` indicates that the stack is not currently in use, so the thread is not running. A stack `size` of `0` indicates that a stack has not been assigned, so the thread has never run.

In Figure 10 the `Telnet` thread is not running, this is because the Telnet service has not been started, see Section 3.9.4. The `Tester` thread has never run because the Tester application has not been started; see Section 3.10.

```
> threads
Debug: 2352/16384, priority=5
RTCManager: 996/16384, priority=1
hardware_watchdog: 740/16384, priority=29
digital_io: 764/16384, priority=29
system_led: 756/16384, priority=25
comms_led: 812/16384, priority=15
zxo_monitor: 868/16384, priority=15
upgrade_firmware: 472/16384, priority=15
ip_ppp_network: 1680/16384, priority=15
net_snmp: 1592/32768, priority=15
Telnet: -1/0, priority=15
Handset: 2700/32768, priority=25
handset_serial_interface: 2048/16384, priority=15
Tester: -1/0, priority=15
utilities_mutex test thread: -1/0, priority=15
vmsSiespacePort0: 11300/16384, priority=15
>
```

Figure 10 - Using `threads`

3.4 FILESTORE COMMANDS

The filestore has a tree structure of files and directories, with directory and file name separated by `/` characters. For example:

```
/var/log
/var/log/system.log
/apps/UTMCVMS.bin
```

File names can be expressed absolutely by ensuring that the name is expressed from the root directory, `/`. Alternatively, filenames expressed without the leading / are relative to the current working directory. GVP also recognises the special directory names:

`.` The current working directory

`..` The parent of the current working directory.

GVP filestore can be navigated and manipulated using commands similar to those provided by other operating systems; `ls`, `cd`, `copy` and `delete` are all available.

The following subsections describe the GVP filestore and go on to describe each of the filestore commands.

3.4.1 The GVP Filestore

The GVP filestore allows data and programs to be stored in files and directories. This filestore offers a similar appearance and structure to the filestore found on the common PC operating systems. However, the GVP resides on Flash memory rather than the disc drives familiar to PC users.

3.4.2 Change the Current Working Directory

The current working directory is the directory in the GVP filestore from which all other relative filenames relate. When the handset interface is started the current working directory is the root directory, `/`. The `cd` command can be used to navigate the filestore.

The synopsis of the `cd` command is:

`cd` *`dir`*

Where:

`dir`	The name of the new directory.

If *`dir`* starts with a `/` then the name is relative to the filestore root. Otherwise, the name is relative to the current working directory.

Figure 11 shows `pwd` and `cd` being used to navigate the GVP filestore.

3.4.3 Display the Current Working Directory

The name of the current working directory can be displayed using `pwd` as follows:

`pwd`

`pwd` displays the directory name, relative to the filestore root, `/`.

Figure 11 shows `pwd` and `cd` being used to navigate the GVP filestore.

```
> pwd
/
> cd/var
> pwd
/var
>
```

Figure 11 - Navigating the GVP Filestore

3.4.4 List Directory Contents

The `ls` command shows the contents of a particular directory. The synopsis of `ls` is:

`ls [`*`dir`*`]`

Where:

`dir` The directory from which the filestore should be displayed.

If the `dir` parameter is not supplied then the contents of the current working directory are displayed. If *`dir`* starts with a `/` then the name is relative to the filestore root. Otherwise, the name is relative to the current working directory.

Figure 12 shows the `ls` command displaying the whole contents of the `/romfs` directory. Each directory is preceded by the `DIR` keyword. Filenames are preceded by their size, measured in bytes. ***The sum of the file sizes is displayed on the last line.***

```
> ls /romfs
DIR            Sep 11 2006 10:29       .
DIR               Sep 11 2006 10:29       ..
  3128      Sep 11 2006 10:29       build.txt
DIR            Sep 11 2006 10:30       pppscripts
106304      Sep 11 2006 10:30       redboot.bin
DIR            Sep 11 2006 10:30       tcl
>
```

Figure 12 - Using `ls`

3.4.5 Display Directory Tree

The `dirtree` command shows the structure of the GVP filestore beneath a particular directory. The whole tree beneath the node is displayed. The synopsis of `dirtree` is:

`dirtree [`*`dir`*`]`

Where:

`dir` The directory from which the filestore should be displayed.

If the `dir` parameter is not supplied then the filestore beneath the filestore root is displayed. If *`dir`* starts with a `/` then the name is relative to the filestore root. Otherwise, the name is relative to the current working directory.

Figure 13 shows the `dirtree` command displaying the whole of the GVP filestore. Each directory is preceded by the `DIR` keyword. Filenames are preceded by their size, measured in bytes. The sum of the file sizes is displayed on the last line.

```
> dirtree
DIR: /etc
DIR: /var
DIR: /var/log
  246097 /var/log/system.log
DIR: /var/queue
DIR: /var/queue/snmpinform
DIR: /var/database
    3002 /var/database/config.xml
    3002 /var/database/config.xml.bak
DIR: /tmp
DIR: /apps
  210004 /apps/UTMCVMS.app
DIR: /romfs
    3128 /romfs/build.txt
DIR: /romfs/pppscripts
     223 /romfs/pppscripts/gprs.script
  106304 /romfs/redboot.bin
DIR: /romfs/tcl
DIR: /dynapps
Total file system usage: 571760
>
```

Figure 13 - Using `dirtree`

3.4.6 Make a Directory

New directories can be created with the `mkdir` command. Typically GVP and GVP applications create the directories they require. So, `mkdir` is unlikely to be useful beyond the development environment.

The synopsis of `mkdir` is:

`mkdir` *`dir`*

Where:

`dir`	The name of the directory to create.

If `dir` is an absolute name then the directory is created beneath the root directory. Otherwise the new directory is created in the current working directory. `dir` cannot contain `/` characters and only one directory can be specified. So, a directory hierarchy cannot be created by a single *`dir`* command.

3.4.7 Delete a File

Files can be removed from the GVP file system using the delete command. The synopsis is:

`delete` *`filename|dir`*

Where:

`filename`	The name of a file to delete
`dir`	The name of a directory to delete.

Absolute or relative filenames can be specified. Only one file or directory can be deleted by each command. Directories are deleted only if they are empty.

> **:** Deleting files could remove critical files from the outstation so that the outstation would not restart after a subsequent reboot. Deleted files cannot be recovered so great care must be taken when deleting files. The command prompts the user before deleting the file.

3.4.8 Display File Contents

The type command writes the file contents to the screen. The synopsis is:

`type` *`filename`*

Where:

`filename`	The name of a file to display.

Relative or absolute filenames can be specified.

The type command dumps the whole file so may take some time to run. There is no way to stop the output once it has started so it must be used with care. Furthermore, it should only be used on text files as binary output may corrupt or lock the terminal output.

3.4.9 Copy a File

GVP files can be copied using the `copy` command. The synopsis is:

`copy` *`srcFile dstFile`*

Where:

`srcFile`	The name of the original file
`dstFile`	The name of the new file.

Relative or absolute filenames can be used in either parameter.

3.5 CONFIGURATION COMMANDS

The configuration of GVP and the GVP applications are held in a configuration file called: `/var/database/config.xml`. The data in this file is modified by the GVP handset commands and by other GVP software.

Some configuration items are set during production, these are not expected to change on any particular installation, and these cannot be changed by a specific handset command. However, GVP provides a handset command that can examine and change any configuration item. Furthermore, the configuration file can be downloaded onto a GVP system, allowing a configuration to be created off-site.

3.5.1 Display and Change a Configuration Item

The configuration items are given a unique key. The keys are divided into sub-keys by a `/` character which gives the keys a structured hierarchy. Configuration can be changed using the `cfgitem` command:

`cfgitem` *`key`* [*`value`*]

Where:

`key`	The unique identifier of the configuration item
`value`	The value to be assigned to the configuration item.

If a `value` is not provided then the current value is displayed.

Figure 14 shows `cfgitem` being used to turn on the Telnet server. This can be done correctly using the services command discussed in Section 3.9.4.

```
> cfgitem ipnetwork/services/telnetserver
item ipnetwork/services/telnetserver is true
> cfgitem ipnetwork/services/telnetserver false
new value for ipnetwork/services/telnetserver set to false
>
```

Figure 14 - Using `cfgitem`

The `cfgitem` command can be used to:

- Develop the factory settings for a particular installation.
- Change values which are normally set in the factory and do not have a GVP handset command to change them.
- Make changes that cannot otherwise be changed due to a problem with GVP firmware.

Where possible, another handset command should be used in preference to `cfgitem` as the official commands check for legal and consistent values.

In some cases, the configuration change takes effect only after a reboot.

3.5.2 Load a Configuration

The GVP `loadcfg` command loads a complete configuration file onto the Gemini^2. The file can be loaded from a TFTP server or through the handset. The synopsis for `loadcfg` is:

`loadcfg handset|[tftp` *`server filename`*`]`

Where:

`handset`	Indicates that the file is to be loaded through the handset
`tftp`	Indicates that the file is to be loaded from a TFTP server
`server`	The IP address of the TFTP server
`filename`	The filename on the server.

To load through a handset requires a terminal emulator that can accept keyboard input from text files. Tera Term Pro, for example, has this facility. Loading via a TFTP server requires a network interface to have been configured.

After loading a new configuration the Gemini^2 is rebooted so that the new configuration is adopted immediately.

This facility is extremely useful for new installations. It is expected that configuration files will be prepared before visiting the site. Consequently, wherever a command changes a configuration item, we describe the items, giving their key, format and default value.

3.5.3 Configure an Application

GVP applications provide their own handset commands to configure, monitor and control their operation. Some GVP applications support the `configure` command. The synopsis of this command is:

`configure` *`applicationName`*

Where:

`applicationName`	The name of an application. See Section 3.10.

This command prompts the handset user to enter configuration values. These prompts collect configuration data for the application and any other GVP modules it uses. This feature can be useful for quickly starting a new configuration. However, the `configure` command is not supported by all GVP applications.

3.6 DATE COMMANDS

GVP provides a command for setting the system clock; configuration items control the time zone and daylight saving calculations.

3.6.1 Display and Change the Date and Time

The `date` command is used to display and change the Outstation's clock. It is important that the clock is set accurately so that data reported to Instations and to log files is correct. Some Instations set the system clock themselves using NTP or SNMP. The Instation update overrides any setting made at the handset interface.

The synopsis of the `date` command is

`date [hh:mm:ss dd/mm/yyyy]|[hh:mm:ss day month year]`

Where:

*hh:*mm*:ss*	The time
dd/mm/yyyy	The date
day month year	The date

The Outstation takes into account whether Daylight Saving Time (DST) is in operation and displays the time as `GMT` or `GMTDST` as appropriate. The time setting values entered should be the local time.

Using `date` without any parameters displays the current date and time, together with the offset, in hours and minutes, from GMT.

```
> date
Tue Sep 22 15:54:55 2009 GMT+1:0 DST

> date 15:55:00 22/09/2009
Tue Sep 22 15:55:00 2009 GMT+1:0 DST
>
```

Figure 15 - Using `date`

3.7 DEBUG COMMANDS

GVP and GVP application software log details of significant events to the GVP Debug Manager. Each software module is named within Debug Manager and the Debug Manager records events with the time, module name and log level.

Event logs are written to the file `/var/log/system.log`. The events which are written depend on the configuration. When modules log an event, they specify a log level. The log level can take one of the values; the highest level is shown first:

```
ERROR
NOTICE
WARNING
INFO
```

Events logged at `ERROR` level indicate that something has gone wrong, `WARNING` logs are less serious and `NOTICE` logs indicate notable events that are not problems. `INFO` level logs show more detailed events or general information about the module.

The `debug` command is used to indicate which log levels are written to file for each module. Only events written to file can be displayed. The `debug` command can also be used to display the log file data. The synopsis of the `debug` command is:

`debug [tail | log | [get all | module] | [set all | module level ]]`

Where:

`tail`	Displays the last few lines of the log file
`log`	Displays the entire log file
`set`	Sets the logging level of the module. All events at this and higher level are logged
`get`	Displays the logging level of the module
`all`	Sets or displays the logging level of all modules
`module`	The name of the module whose log level is to be set or displayed
`level`	The log level to be applied to the module. This is only used with `set`. The `level` can be set to: `none`, `error`, `warning`, `notice` or `info`. This level and all higher-level events from the module are written to the file and can be displayed.

Using `debug` without parameters gives a live display of events. Each event is displayed as it is logged. The output stops when any key is pressed.

The names of all available modules can be obtained using `debug get all`.

By default, all modules log at `NOTICE` level. Logging can slow the operation of the Gemini^2 and contribute to wear on the FLASH memory. So, the log level should be returned to the default level when the investigations are complete.

3.8 INTERFACE COMMANDS

Gemini^2 systems can have several serial and digital interfaces. The precise configuration depends on the expansion cards that are attached to the Gemini^2 Processor card.

Typically, the interfaces are configured and manipulated by GVP applications using functions provided by GVP. However, GVP does provide some handset commands for monitoring and manipulating the Gemini^2 interfaces in support situations.

3.8.1 Interface Names

The interface names presented by GVP depend on the number, type and address of the expansion cards attached to the Gemini^2 Processor card. In general, the following interfaces are available:

`zxo.in.1`	A ZXO (Zero Cross-Over) interface. This is connected to the PSU. The GVP uses this interface to detect mains failure
`cpu.out.1`	Relays mounted on the Gemini^2 Processor card, marked RL1 and RL2
`busmova`*`n`*`.in.1`	Digital Inputs on the n^{th} BUS/MOVA card

`busmova`*`n`*`.out.1`	Digital Outputs on the n^{th} BUS/MOVA card
`iocard`*`n`*`.in.1`	Digital Inputs on the n^{th} BUS/MOVA card
`iocard`*`n`*`.out.1`	Digital Outputs on the n^{th} generic I/O card
`iocard`*`n`*`.mon`	Voltage Monitors on the n^{th} generic I/O card
`iocard`*`n`*`.dio.`*`m`*	The m^{th} Digital IO interface on the n^{th} generic I/O card
`iocard`*`n`*`.dio.ctrl`	A control register used to assign the direction of the Digital IO interfaces.

3.8.2 Display Cards

The expansion cards attached to the Gemini^2 Processor card can be listed using the `cards` command. The synopsis is:

```
cards
```

The output lists each attached card and:

`name`	A unique name for the card within the Gemini^2
`address`	A software address for the card
`type`	The type of card

```
> cards
iocard1: address=0x74000000, generic iocard
busmova2: address=0x75000000, bus mova comms iocard
>
>
```

Figure 16 - Using `cards`

3.8.3 Display Digital Inputs

The state of the digital input devices can be monitored using the `din` command. The synopsis of `din` is:

```
din [device]
```

Where:

`device`	The digital input device to monitor.

Without parameters, `din` lists the available digital input devices. Otherwise, the state of the named device is shown. Each digital input is represented by a binary digit (bit). Each device supports up to 48 bits. The `din` output shows the combined bit values for the whole device each time that an input state changes. The output stops after any key is pressed.

Note: To display input lines in binary (1 or 0) use the WIZ command, see section 3.2.

Figure 17 shows `din` being used to list the devices and monitor the `iocard1.in.1` device. The device is attached to a single induction loop on bit 0 (`0x01`) and a bi-

directional pair of loops on bits 2 and 3 (`0x04` and `0x08`) used by the UTMC Car Park application. We can see:

`0x0 0x1 0x0`	A vehicle crossing the single loop
`0x0 0x4 0xc` `0x8 0x0`	A vehicle crossing the dual loops.

The `din` command can be used to test the connection and operation of devices connected to the Gemini[2].

```
> din
usage: din [InputName]

Current registered inputs:

zxo.in.1
iocard1.in.1
iocard1.vmon
iocard1.dio.1
iocard1.dio.2
iocard1.dio.3

> din iocard1.in.1

Monitoring of iocard1.in.1 started, press any key to exit...

Time: Fri Jan 02 05:24:33 1970
New value: 0x0

Time: Fri Jan 02 05:24:55 1970
New value: 0x1

Time: Fri Jan 02 05:24:57 1970
New value: 0x0

Time: Fri Jan 02 05:24:59 1970
New value: 0x4

Time: Fri Jan 02 05:24:59 1970
New value: 0xc

Time: Fri Jan 02 05:25:00 1970
New value: 0x8

Time: Fri Jan 02 05:25:00 1970
New value: 0x0

>
```

Figure 17 - Using `din`

3.8.4 Set Digital Outputs

The state of the digital output devices can be controlled using the `dout` command. Outputs can only be set on an unconfigured outstation or where output bits are not being controlled by the application. The synopsis of `dout` is:

`dout [`*`device bit value`*`]`

Where:

device	The digital output device to control
bit	The device bit to control
value	The value to assign to the *bit*. This can be 0 or 1.

Without parameters dout lists the available digital output devices together with the current output data value. With parameters, the state of the named device is set. Each digital output is represented by a binary digit (bit). Each device supports up to 64 bits.

The dout command can be used to test the connection and operation of devices connected to the Gemini^2.

```
> dout iocard1.out.1
usage: dout [OutputName bit value]

Current registered outputs:

cpu.out.1
iocard1.out.1
iocard1.dio.1
iocard1.dio.2
iocard1.dio.3
iocard1.dio.ctrl

> dout iocard1.out.1 0 1
Output bit 0 set to 1
> dout iocard1.out.1 47 1
Output bit 47 set to 1
>
```

Figure 18 - Using dout

3.8.5 Controlling Modem Power

In some hardware configurations, the Gemini^2 provides the power for attached modems and other devices. Typically the GVP turns the modem power on and off as required. Sometimes, it may be necessary to cycle the modem power in order to force a disconnection or clear some other condition.

The modem power can be turned on and off using the modempower command as follows:

```
modempower on|off
```

Where:

on	Turns the power on
off	Turns the power off.

 : Use with caution. Turning the modem power off remotely will cause communications to the outstation to be lost and a site visit will be required.

3.9 NETWORK COMMANDS

There are several GVP handset commands to describe how the Gemini^2 attaches to the available networks and the parameters to use with particular protocols.

3.9.1 Configure Network Settings

Network interfaces are configured using the `network` command. Gemini^2 has two network interfaces, an Ethernet port (`eth0`) and a serial port (`ppp0`). Configuration of these two ports is very different so the `network` command has two forms. The synopsis is:

```
network eth0 down|bootp|[manual [ipaddress [netmask [broadcast [gateway]]]]]

network ppp0 down|up|[setup [script [username [passwd [all|chap|pap]]]]]
```

Where the parameters are:

`eth0`	The name of a network interface
`down`	Stops the network interface. After using this parameter the network interface cannot be used
`bootp`	Configures the interface to use the BOOTP protocol to obtain its IP address. At present, this option is not supported
`manual`	Indicates that the network parameters should be taken from the remaining parameters
`ipaddress`	The IP address to be assigned to the network interface. This should be a given in dotted decimal notation, e.g. 192.168.1.1
`netmask`	The network mask for the interface. If this parameter is not specified then a value based on the *`ipaddress`* is chosen. The network mask must be specified in dotted decimal notation
`broadcast`	The broadcast address for the attached network. This parameter is required if the Gemini^2 implements any broadcast protocols. The broadcast address must be specified in dotted decimal notation
`gateway`	The IP address of the default gateway. This must be specified if the Gemini^2 communicates with other systems through a router; this parameter specifies the IP address of the router. The gateway address must be specified in dotted decimal notation
`ppp0`	The name of a network interface
`up`	Brings up network interface
`setup`	The remaining parameters configure the network interface
`script`	The name of a script file. The file contains a chat script for logging on the PPP interface
`username`	The username required to logon to the remote network
`passwd`	The password used to logon to the remote network
`chap`	GVP should use CHAP protocol to authenticate itself to the network
`pap`	GVP should use the PAP protocol to authenticate itself to the

network

`all` The network decides the authentication procedure, CHAP or PAP.

Using the network command without parameters shows the current network configuration and status. The meaning of the status flags is covered in Section 3.9.3.

Using `eth0` specifies the parameters for the Gemini^2 Ethernet port.

Using `ppp0` specifies the Point-to-Point Protocol (PPP) parameters for the modem port. The PPP is used to connect the Gemini^2 to an IP network. Typically, the Gemini^2 is connected through modems, possibly via a mobile network. The chat script is used to instruct the modems to autodial and the authentication protocol transfers the username and password to the network access point. The network provides the Gemini^2's IP address and other parameters using PPP.

Figure 19 demonstrates the `network` command being used to specify the parameters for `eth0` and then showing the current configuration. Note that a configuration is specified for the `lo0` interface. This is the loopback interface; it is used by the Gemini^2 to make connections to itself. The `lo0` parameters cannot be modified by handset commands.

```
> network eth0 manual 192.168.1.1
> network
eth0    Flags UP BROADCAST RUNNING SIMPLEX MULTICAST
        Address 192.168.1.1 Mask 255.255.0.0 Broadcast 255.255.255.255

lo0     Flags UP LOOPBACK RUNNING MULTICAST
        Address 127.0.0.1 Mask 255.0.0.0
>
```

Figure 19 - Using the `network` command

3.9.2 Ping

GVP provides a `ping` command to test network communications. This is similar to the `ping` command provided on other platforms. The synopsis is:

`ping [`*`ipAddress`*`|`*`hostname`*`]`

Where:

`ipAddress` The IP address of the system to test
`hostname` The DNS host name of the system.

Figure 20 demonstrates the use of `ping` to show that messages can be exchanged with the IP address 137.223.72.84. The `ping` command repeatedly sends ICMP Echo Requests and calculates the response time. `ping` stops when any key is pressed.

A response from the target system indicates that there is a network route between the Outstation and the target, and that the target system is running. It does not indicate that the Outstation and target can communicate using other protocols; both systems must

be running the appropriate software and access by these protocols must not be barred by intervening firewalls.

```
> ping 137.223.72.84
ping 137.223.72.84: num: 0, sent: 1, ok: 1, avg delay: 0ms
ping 137.223.72.84: num: 1, sent: 1, ok: 1, avg delay: 0ms
ping 137.223.72.84: num: 2, sent: 1, ok: 1, avg delay: 0ms
ping 137.223.72.84: num: 3, sent: 1, ok: 1, avg delay: 0ms
ping 137.223.72.84: num: 4, sent: 1, ok: 1, avg delay: 0ms
ping 137.223.72.84: num: 5, sent: 1, ok: 1, avg delay: 0ms
ping 137.223.72.84: num: 6, sent: 1, ok: 1, avg delay: 0ms
ping 137.223.72.84: num: 7, sent: 1, ok: 1, avg delay: 0ms
ping 137.223.72.84: num: 8, sent: 1, ok: 1, avg delay: 0ms
ping 137.223.72.84: num: 9, sent: 1, ok: 1, avg delay: 0ms
>
```

Figure 20 - Using `ping`

3.9.3 Display Routing Tables and Statistics

The network routing table describes how IP packets are transmitted from the Gemini^2 in order to reach destinations on different networks. The `route` command displays the contents of the routing table and displays some statistics about the usage of the network interfaces. The synopsis of the route command is:

```
route
```

Figure 21 shows example output from `route`. We see that the output is divided into a routing table and interface statistics.

3.9.3.1 Routing Table

The routing table shows how packets are sent from the Gemini^2. In particular it shows the transmission interface and the next device to receive the packet. The routing table can show problems in the Gemini^2 `network` configuration. In particular, it can show that some IP addresses cannot be reached by the Outstation.

When a packet is to be transmitted, GVP examines the packet's destination IP address and searches through each row of the routing table. It applies the subnet `Mask` from the routing table to the packet's IP address. It then applies the same `Mask` to the `Destination` address from the routing table entry. If the two results are the same then the row could be used to route the packet. The route chosen is the one with largest `Mask`; the one with the smallest number of trailing zeros.

Example
Using the routing table shown in Figure 21, this is how an IP packet with IP address `137.223.73.84` is routed:

1. The routing software examines the first row of the routing table. It masks the packet's IP address, `137.223.73.84`, with `Mask 0.0.0.0` to give `0.0.0.0`. It then masks the row's IP address, `0.0.0.0`, with `0.0.0.0` to give `0.0.0.0`. Both calculations give `0.0.0.0` so this is a candidate route.

2. Using the second row, `137.223.73.84` is masked with `255.0.0.0` giving `137.0.0.0`. Masking the table data, `127.0.0.0` and `255.0.0.0` gives `127.0.0.0`. So the row cannot be used.
3. Row 3 does not have a mask so the addresses are not masked. As `137.223.73.84` does not match `127.0.0.1`, row 3 is not used.
4. With row 4, `137.223.73.84` is masked with `255.255.0.0` giving `137.223.0.0`; `137.223.0.0` is masked with `255.255.0.0` giving `137.223.0.0`. So row 4 can be used.
5. This process has found that rows 1 and 4 are candidate routes. Row 1 produced a masked IP address of `0.0.0.0` and row 4 produced `137.223.0.0`. Row 4 is chosen as its result has the smallest number of trailing zeros in the mask.

Once a route has been decided, the packet is transmitted on the interface identified by the `Interface` field of the routing table. If the `Flags` field contains a `G` (gateway) character then the packet will be sent to a router; the router's IP address is given in the `Gateway` field of the routing table.

```
> route
Routing tables
Destination     Gateway         Mask            Flags    Interface
0.0.0.0         137.223.83.1    0.0.0.0         UG       eth0
127.0.0.0       127.0.0.1       255.0.0.0       UG       lo0
127.0.0.1       127.0.0.1                       UH       lo0
137.223.0.0     137.223.0.0     255.255.0.0     U        eth0
Interface statistics
eth0    IP: 137.223.152.216, Broadcast: 137.223.255.255, Netmask: 255.255.0.0
UP BROADCAST RUNNING MULTICAST MTU: 1500, Metric: 0
Rx - Packets: 59, Bytes: 6383, Tx - Packets: 45, Bytes: 3965
lo0     IP: 127.0.0.1, Broadcast: 127.0.0.1, Netmask: 255.0.0.0
UP LOOPBACK RUNNING MULTICAST MTU: 16384, Metric: 0
Rx - Packets: 0, Bytes: 0, Tx - Packets: 0, Bytes: 0
>
```

Figure 21 - Using `route`

The columns of the routing table are:

`Destination` — Used to find a routing table entry for a destination address. If the masked IP address matches this value then this route can be used for the packet

`Gateway` — If the `Flags` include a `G` then this is the IP address of a router. Otherwise, it is the masked IP address of the interface

`Mask` — The network mask of the interface

`Flags` — Describe the table entry:

- `U` — Indicates that the interface is up. Ethernet interfaces are always up if they are defined
- `G` — Packets are sent via a router
- `H` — The route is direct to a host. The `Mask` is not used
- `D` — The router created the route
- `M` — The route was created by the Gemini^2 but was modified by the router

`Interface` — The interface on which packets should be transmitted.

The routing table entries are created from the parameters of the network command (see Section 3.9.1). The entry with `destination 0.0.0.0` is only created if a `gateway` parameter is supplied. This is known as the default route; it indicates that any packets with non-local addresses should be sent to the router. Configurations without a gateway can only send data to devices on the local network.

3.9.3.2 Interface Statistics

The `route` command displays configuration, status and statistical data about each network interface; see Figure 21 for an example. For each interface, route displays three rows of data.

The first row displays configuration data, this is the data entered by the network command; see Section 3.9.1:

name	The name of the interface
IP:	The IP address of the interface
Broadcast:	The broadcast IP address of the interface
Netmask:	The network mask of the interface.

The second row shows the status of the interface:

UP	Indicates that the interface is operational
LOOPBACK	This is the loop-back device, used when the Outstation sends packets to itself
BROADCAST	Indicates that all devices attached to the interface receive the transmitted packets. This is always set for Ethernet Interfaces
SIMPLEX	Indicates that the interface cannot read the packets it has transmitted
RUNNING	Indicates that the interface has all its required resources
MULTICAST	Indicates that the interface can multicast
MTU	The Maximum Transmission Unit, the maximum size of link level packets on the interface
Metric	An indication of the cost of using this interface. This may affect the routing decision. This value is always 0.

The third row shows statistics for the interface:

Rx - Packets	The number of received packets
Rx – Bytes	The number of received bytes
Tx - Packets	The number of transmitted packets
Tx – Bytes	The number of transmitted bytes.

3.9.4 Start Network services

GVP supports the Telnet Protocol and the Trivial File Transfer Protocol (TFTP).

Telnet allows another system on the network to access the GVP handset interface and use the handset commands described in this document. This means that the Outstation can be monitored and controlled from another part of the network.

TFTP is a protocol for transferring files. This can be extremely useful as files can be exchanged with an Instation system or with an engineer's notebook computer. TFTP can be used to deliver new program or configuration files or collect debug log files.

GVP includes servers for both TFTP and Telnet. However, they are initially disabled. The `services` command can be used to start and stop these servers. The synopsis of the `services` command is:

```
services TFTP|TELNET [start|stop]
```

Where:

`TFTP`	Indicates that the TFTP server is to be started or stopped
`TELNET`	Indicates that the Telnet server is to be started or stopped
`start`	Start the server
`stop`	Stop the server.

Calling `services` without the `start` or `stop` parameters displays the current state of the service.

TFTP is an insecure protocol; servers allow file access without even a password check. Many Outstations are configured without a password so Telnet can also have security problems. So, these servers should only be started when necessary. Telnet should only be started if remote access to the handset commands is required. It is recommended that TFTP be started only when a file is to be transferred.

3.9.5 SNMP

GVP systems can be managed using the SNMP. Furthermore, some GVP applications use SNMP to communicate with Instations. GVP provides handset commands for configuring and testing SNMP.

3.9.5.1 Set SNMP Community

GVP Outstations can use SNMP to communicate with Instations if both are members of the same SNMP community. This is a weak security mechanism specified by SNMP. GVP systems can be the member of one SNMP community, as specified by the `snmpcommunity` command:

```
snmpcommunity [community]
```

Where:

`community`	The community to which the Instation belongs.

Using `snmpcommunity` without parameters displays the current community name.

The community name must be agreed with the administrators of the Instation.

```
> snmpcommunity UTMC
Community: UTMC
> snmpcommunity
Community: UTMC
>
```

Figure 22 - Using `snmpcommunity`

Figure 22 shows the SNMP community name being set to UTMC. This name is required by most applications that implement UTMC SNMP MIBs.

3.9.5.2 Configure SNMP Security

SNMP Version 3 (SNMPv3) offers a higher level of security than that supported by its predecessors. Along with the standard SNMP data, a username and a calculated checksum, or hash, are transmitted. The username identifies the remote system and is used to determine its access rights. The hash is used to confirm that the data has not been modified and confirms the source of the data.

The hash is a multi-byte value calculated by processing the SNMP data, username and password through a hashing algorithm. SNMPv3 supports two hashing algorithms MD5 and SHA-1. The hash is transmitted along with the SNMP message and username; the password is not transmitted. The receiver processes the message, username and password to confirm the hash.

The username and password must be agreed between the Instation and Outstation. The authenticity and integrity of the data relies on the password being kept a secret, shared between the Instation and Outstation. The confidentiality of the data is not assured as the SNMP data is sent as plain text.

The hashing algorithm is configured with the `snmpuser` command. The synopsis is:

`snmpuser NONE|[MD5|SHA` *`username password`*`]`

Where:

`NONE`	Hashing is not performed
`MD5`	MD5 hashing protects messages
`SHA1`	SHA-1 hashing protects messages
`username`	Identifies the remote system and its access rights
`password`	The key to the hashing algorithm

3.9.5.3 Configure SNMP Traps

In general, GVP responds to SNMP requests from Instations. Under some circumstances, GVP may need to send unsolicited SNMP messages to the Instation; these messages are SNMP Traps or SNMP Inform Requests.

The `snmptrap` command is used to specify the type of unsolicited message sent by the Gemini^2 and also specifies the address of the Instation to receive these messages. The synopsis of `snmptrap` is:

```
snmptrap [v1TRAP|v2cTRAP|v2cINFORM|v3TRAP|v3INFORM ipaddress]
```

Where the parameters are:

`v1TRAP`	Indicates that SNMP Traps are transmitted from GVP, as defined by SNMP version 1
`v2cTRAP`	Indicates that SNMP Traps are transmitted from GVP, as defined by SNMP version 2
`v2cINFORM`	Indicates that SNMP Inform Requests are transmitted from GVP, as defined by SNMP version 2
`v3TRAP`	Indicates that SNMP Traps are transmitted from GVP, as defined by SNMP version 3
`v3INFORM`	Indicates that SNMP Inform Requests are transmitted from GVP, as defined by SNMP version 3
`ipAddress`	The IP address of the Instation. The address must be specified in dotted decimal notation.

The Gemini^2 can use only one form of Trap or Inform Request; this must be agreed with the administrators of the Instation. However, GVP applications may specify an alternative destination for their unsolicited messages.

3.9.5.4 Test SNMP Traps

The `trap` command sends a single SNMP Trap or Notify Request. The format and destination of the request is specified by the `snmptrap` and `snmpcommunity` commands (see Sections 3.9.5.1 and 3.9.5.2). The format of the `trap` command is:

```
trap
```

The trap is sent as configured by the `snmptrap` command, See Section 3.9.5.3. The `trap` command can be used to check the configuration of the Gemini^2, network and Instation.

3.9.5.5 Configure SNMP

The `snmpsetup` command can be used to specify values for MIB-II objects. Instations can use SNMP to read these values from the GVP to identify the Outstation.

The synopsis of `snmpsetup` is:

```
snmpsetup [SysLocation [SysName [SysContact]]]
```

Where the parameters specify values for the MIB-II objects:

`SysLocation`	The physical location of the Gemini^2
`SysName`	The name of the Gemini^2. Administrators of the Instation may have specified this

SysContact Contact details of those responsible for the Outstation.

Using snmpsetup without parameters displays the current object values as well as a value for sysDescr; this describes the GVP software.

```
> snmpsetup "York Rd, Poole" "York Road" "0771 555 989"
> snmpsetup
System Location: York Rd, Poole
System Name: York Road
System Contact: 0771 555 989
System Description: 667/TZ/31760/000 issue 3.0 (3 ) (build 4) RELEASE  :
REL_GVP_KERNEL_5 : REL_REDBOOT_9
>
```

Figure 23 - Using snmpsetup

Figure 23 shows snmpsetup being used to set and display object values.

3.9.5.6 Show SNMP MIBS

Instations monitor and control some GVP applications using SNMP. The data objects accessed by Instations is organised into groups of MIBs. GVP applications that support SNMP typically provide data for one MIB; these applications register their support for a MIB with GVP.

The snmpmibs command displays those MIBs that are being supported by the running applications. The synopsis is:

```
snmpmibs
```

Figure 24 shows that a GVP system is running a GVP application that supports the UTMC VMS MIB.

```
> snmpmibs
List of registered mibs:
VMS
>
```

Figure 24 - Using snmpmibs

3.9.6 NTP

GVP Outstations can synchronise their system clocks with Instation clocks. This is done using the Network Time Protocol (NTP). An Instation system should be configured as an NTP server and Outstations should be configured as an NTP client - see 4.7.3.

The use of NTP is optional. It allows all Instations and Outstations in a network to be synchronised; offering benefits such as consistent log data.

The timeserver command displays the NTP port and server address. It also displays elapsed time since the last NTP time update and the computed accuracy.

3.9.6.1 Configure SNTP Server Address

[This command is not used on GVP versions 11 and above. SNTP is replaced by NTP].

The address of the SNTP server to be used by an Outstation is set using the `sntpserver` command. The synopsis is:

`sntpserver [`*`ipaddress`*`]`

Where:

`ipAddress`	The IP address of the Instation that provides the timeserver. This should be in dotted decimal notation.

When used without parameters this command shows the currently used IP address.

3.9.6.2 Set SNTP Synchronisation Time

[This command is not used on GVP versions 11 and above. SNTP is replaced by NTP].

The interval at which an Outstation attempts to resynchronise with the Instation is set using the `sntpsync` command. The synopsis is:

`sntpsync [`*`timeInSeconds`*`]`

Where:

`timeInSeconds`	The synchronisation period in seconds

When used without parameters this command shows the currently used synchronisation interval.

3.9.7 DNS

The Domain Name System (DNS) is the network of computer systems that enables a host name such as `myGemini` or `www.siemens.com` to be translated to an IP address, such as `192.138.228.1`.

DNS domains are the part of the hostname after the first "." character. For example, the host name `productdev.stcl.siemens.co.uk` is in the `stcl.siemens.co.uk` domain. In turn, the domain `stcl.siemens.co.uk` is in the `siemens.co.uk` domain. This gives a hierarchy of domains, `uk` being a top-level domain.

In some cases, when IP addresses are supplied to GVP, DNS hostnames can be provided instead. This can make the configuration of GVP easier and less prone to errors. It also allows the network administrators to change IP addresses without having to reconfigure lots of systems.

To use DNS, the network must provide a host, which provides a DNS server. The GVP Outstation must be configured with the IP address of the Outstation.

3.9.7.1 Configure DNS

The `ns` command configures GVP to use DNS. The synopsis of `ns` is:

`ns [`*`nsIpAddress`* `[`*`domainName`*`]]`

Where:

`nsIpAddress`	The IP address of the DNS server
`domainName`	The default domain name.

The default domain name is appended to any host names that do not include a domain name. Figure 25 shows `ns` being used to define a DNS server and a default domain of `stcl.siemens.co.uk`. It shows that `productdev` and `productdev.stcl.siemens.co.uk` resolve to the same IP address.

```
> ns 137.223.44.131 stcl.siemens.co.uk
> host productdev
DNS lookup: productdev -> 137.223.152.130
> host productdev.stcl.siemens.co.uk
DNS lookup: productdev.stcl.siemens.co.uk -> 137.223.152.130
>
```

Figure 25 - Using `ns` and `host`

Using `ns` without any parameters shows the current configuration.

3.9.7.2 Translate Hostnames

The `host` command translates a DNS host name to an IP address. The `host` command can be used to test the DNS configuration and to obtain IP addresses for GVP commands, which do not accept hostnames.

The synopsis of the `host` command is:

`host` *`hostname`*

Where:

`hostname`	The hostname to be translated.

Examples of `host` being used can be found in Figure 25.

3.9.8 DDNS

Typically, the DNS servers introduced in Section 3.9.7 have their database configured by administrators who set up the name to address translations. Some DNS servers can be configured using the Dynamic DNS protocol (DDNS).

Typically, DDNS clients contact the DDNS server with their hostname and IP address. The DNS provides this information to other systems wishing to contact the DDNS client. DDNS is not always support by DNS servers. However, it can offer greater flexibility in

network configurations. It is particularly useful in situations where the IP addresses can change; some GPRS networks reassign IP addresses often.

3.9.8.1 Configure DDNS

The `ddns` command configures GVP as a DDNS client. The synopsis is:

`ddns [disable|enable|[setup` *`nameserver portnumber hostname domainname`*`]]`

Where:

`disable`	The GVP DDNS client is disabled. It stops contacting the server
`enable`	The GVP DDNS client is enabled. It updates the server when required
`setup`	Subsequent parameters configure the GVP DDNS client
`nameserver`	The IP address of the DDNS server
`portnumber`	The UDP port number of the DDNS server
`hostname`	The host name of the Gemini^2, without the domain portion
`domainname`	The DNS domain name of the Gemini^2.

Using `ddns` without parameters displays the current configuration. To activate a DDNS configuration the `setup` and then the `enable` parameters must be used. Figure 26 shows the DDNS configuration being created and tested.

At present, BOOTP is not supported by GVP; the IP address on `eth0` can only be set manually. Consequently, the DDNS client only reports IP address changes made on the `ppp0` interface.

3.9.8.2 Test DDNS

The `ddnstest` command performs a DDNS update with the server. The synopsis is:

`ddns` *`interface`*

Where:

`interface`	The network interface through which to contact the server.

The output from `ddnstest` indicates whether the update was successful. Figure 26 shows `ddns` and `ddnstest` being used to update the configuration and server; the `host` command is used to confirm that the update has taken place.

```
> dns setup 137.223.152.130 2304 spud gemini.stcl.siemens.co.uk
> ddns
DDNS Client: Disabled
DDNS server: 137.223.152.130
DDNS port number: 2304
Host Name: spud
Domain Name: gemini.stcl.siemens.co.uk
> ddns enable
> ddnstest eth0
ddns performed correctly
```

```
> host spud
DNS lookup: spud -> 137.223.152.216
>
```

Figure 26 - Using and Testing `ddns`

Note that `ddnstest` works on `eth0`, even though the DDNS client does not.

3.10 APPLICATION CONTROL COMMANDS

The Gemini^2 Processor cards are distributed with the GVP software installed in Flash memory. The Flash memory also contains one or more GVP applications. Different Processor cards are installed with different applications, depending on how the Processor card will be used. The applications are started and stopped using the `start` and `stop` commands.

Typically, GVP applications provide their own handset commands. These are documented in the handbook for that application.

3.10.1 Starting GVP Applications

GVP applications can be started using the `start` command. Its synopsis is:

`start` *`applicationName`*`|[list [a`*`pplicationName`*`]]`

Where:

`applicationName`	The name of an application. Case is not significant
`list`	Indicates that the state of the application is required.

When the `list` keyword is used then the state of the named application is displayed. If no name is given then the state of all installed applications is displayed. Using `start` without the `list` keyword starts the named application. The application will be started on subsequent reboots.

```
> start list
Application list:

Tester V0.0.0                           Stopped
UTMCVMS V1.0.1                          Stopped
> start utmcvms
Application utmcvms started
> start list
Application list:

Tester V0.0.0                           Stopped
UTMCVMS V1.0.1                          Started
>
```

Figure 27 - Using `start`

Figure 27 shows `start` being used to determine the status of the UTMC VMS application and then starting it.

3.10.2 Stopping GVP Applications

 : Use with caution. Stopping an application will affect the outstation functionality.

GVP applications can be stopped using the `stop` command. Its synopsis is:

`stop` *`applicationName`*

Where:

`applicationName` The name of an application. Case is not significant.

This stops the named application. The application will not be started on subsequent reboots.

The `start` and `stop` commands modify the following configuration items:

3.11 UPGRADING

The Gemini^2 firmware consists of:

Bootstrap Loader	Provides facilities for installing and starting
Launcher	Controls the initial switching between the Bootstrap Loader and GVP.
GVP	Provides basic and general-purpose functionality such as filestore management, debug logging and network interfaces. GVP can start GVP applications
GVP applications	Provides business functionality. This varies with the application. Examples include monitoring car park occupancy and monitoring remote cabinets.

The `upgrade` command is used to upgrade the boot loader component. Upgrade of any other components should be performed via the web interface – see section 4.3.

If GVP is not already running, then the GeminiUpgrade Tool can be used to install a complete flash image. This would be used for the first-time installation of GVP.

The synopsis of the upgrade command is:

`upgrade [`*`server file`*`] | [bootloader [`*`file`*`]]`

3.11.1 Upgrading the Bootstrap Loader

The system engineer must be on-site to upgrade the bootstrap loader.

If the upgrade of the bootstrap loader fails then the Gemini^2 Processor card can become unusable; it will have to be returned to the factory for repair. Consequently, the upgrade cannot be done over a network. The upgrade file must be resident on the GVP filestore before the process is started. As an additional precaution, it is recommended that a serial-port handset be used rather than Telnet; this removes the risks associated with a temporary network problem.

The file can be delivered to the GVP filestore using TFTP. More typically, it is written to the GVP filestore during a GVP upgrade.

Before the upgrade, the Gemini^2 watchdog must be disabled by removing link PL6 link on the Gemini^2 Processor card.

To upgrade the bootstrap loader, use the `upgrade` command as follows:

`upgrade bootloader [`*`file`*`]`

Where:

`bootloader`	Indicates that the bootstrap loader should be upgraded
`file`	The file containing the new bootstrap loader.

Upgrading the bootstrap loader takes less than one second. After the upgrade, replace the watchdog link and reboot the Gemini^2.

3.12 MISCELLANEOUS COMMANDS

This section describes the GVP commands which have been difficult to classify so are not described elsewhere.

3.12.1 Set Password

The GVP password can be set or changed using the `passwd` command. The synopsis is

`passwd` *`newPassword`*

Where:

`newPassword`	The new password.

After using `passwd`, subsequent attempts to start a handset dialog require the password to be entered. This applies to both handset and Telnet sessions.

3.12.2 Watching Other Commands

The `watch` command runs other commands periodically. The synopsis of `watch` is

`watch` *`time command`* [*`parameter`*`...]`

Where:

time	The number of seconds between consecutive calls to the command
command	The command to execute
parameter	A command parameter.

The `watch` command repeatedly calls the command until a key is pressed.

This command is useful when waiting for a particular condition or for monitoring the use of system resources such as Processor loading, memory or filestore usage. Figure 28 shows `watch` being used to monitor Processor loading.

```
>  watch 1 cpuload
0.1 Second average: 0%
1 Second average:   7%
10 Second average:  13%

0.1 Second average: 0%
1 Second average:   2%
10 Second average:  12%

0.1 Second average: 0%
1 Second average:   0%
10 Second average:  11%

0.1 Second average: 0%
1 Second average:   0%
10 Second average:  9%

0.1 Second average: 0%
1 Second average:   1%
10 Second average:  9%

>
```

Figure 28 - Using `watch`

3.12.3 Hello World

The `helloworld` program displays the command line parameters. This is used in development to test the handset interface. It is unlikely to be generally useful.

The synopsis of the `helloworld` command is:

`helloworld` *`parameter...`*

Where:

parameter	Any text.

3.12.4 Exit

The exit command closes the current Telnet session. The command has no effect when used on a directly connected handset terminal. The synopsis is:

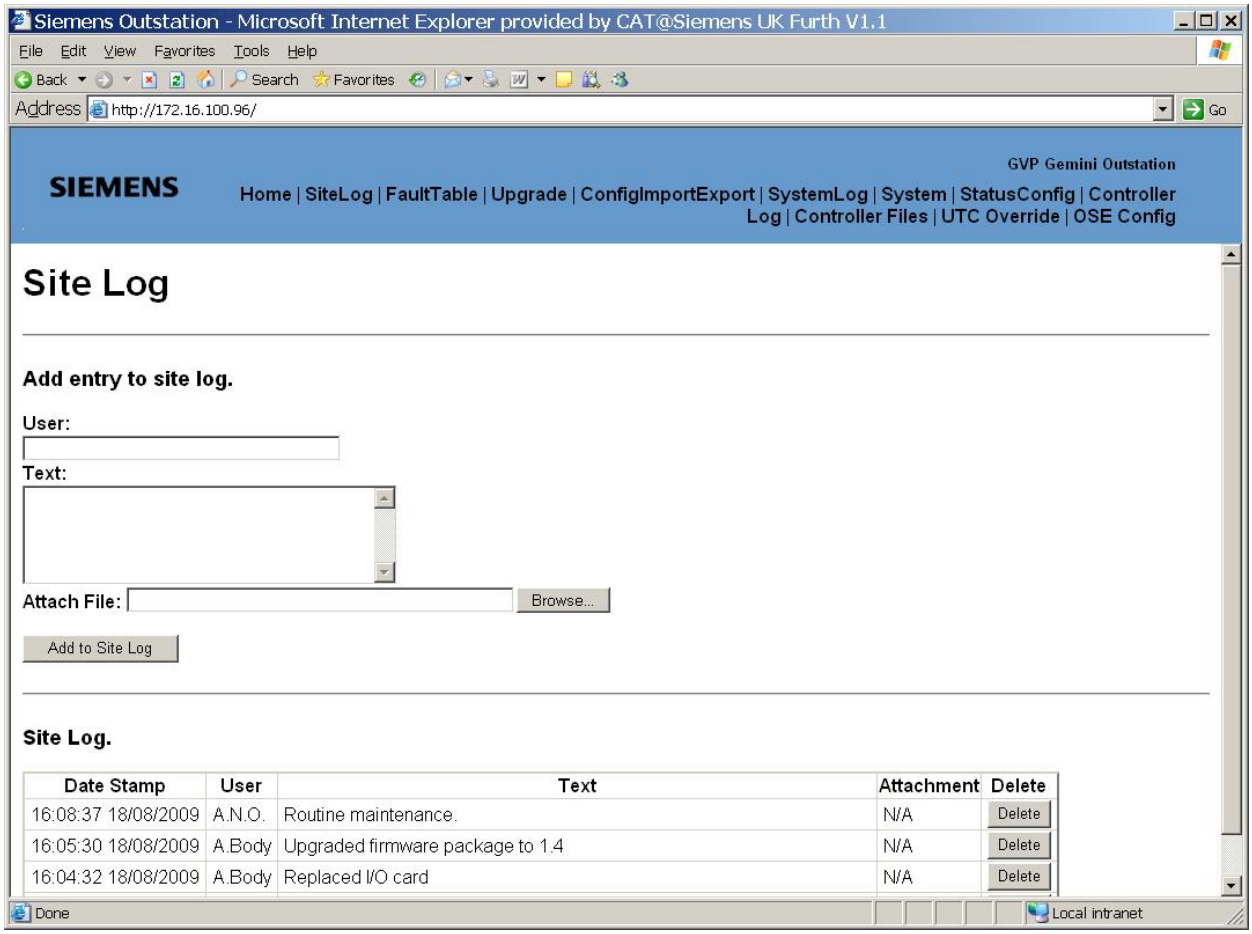
`exit`

4. UTMC Common Web Interface Screens

The outstation provides the following common web screens. Additional screens can be added by each application.

4.1 SITE LOG SCREEN

This screen allows the entry any display of maintenance operations which have been carried out on the outstation and associated street equipment e.g. fault fixes, re-configurations, upgrades etc.



User:

Enter the name of the user who is making the log entry.

Text:

Type the text to be entered into the log.

Browse

If attaching a file to the log entry, press ‘Browse’ and then navigate to the file to be attached. Large files (more than 500K) should be avoided. The available space is dependent on the outstation application. This can be checked using the ‘diskusage’ command on the terminal interface.

Add to Site Log

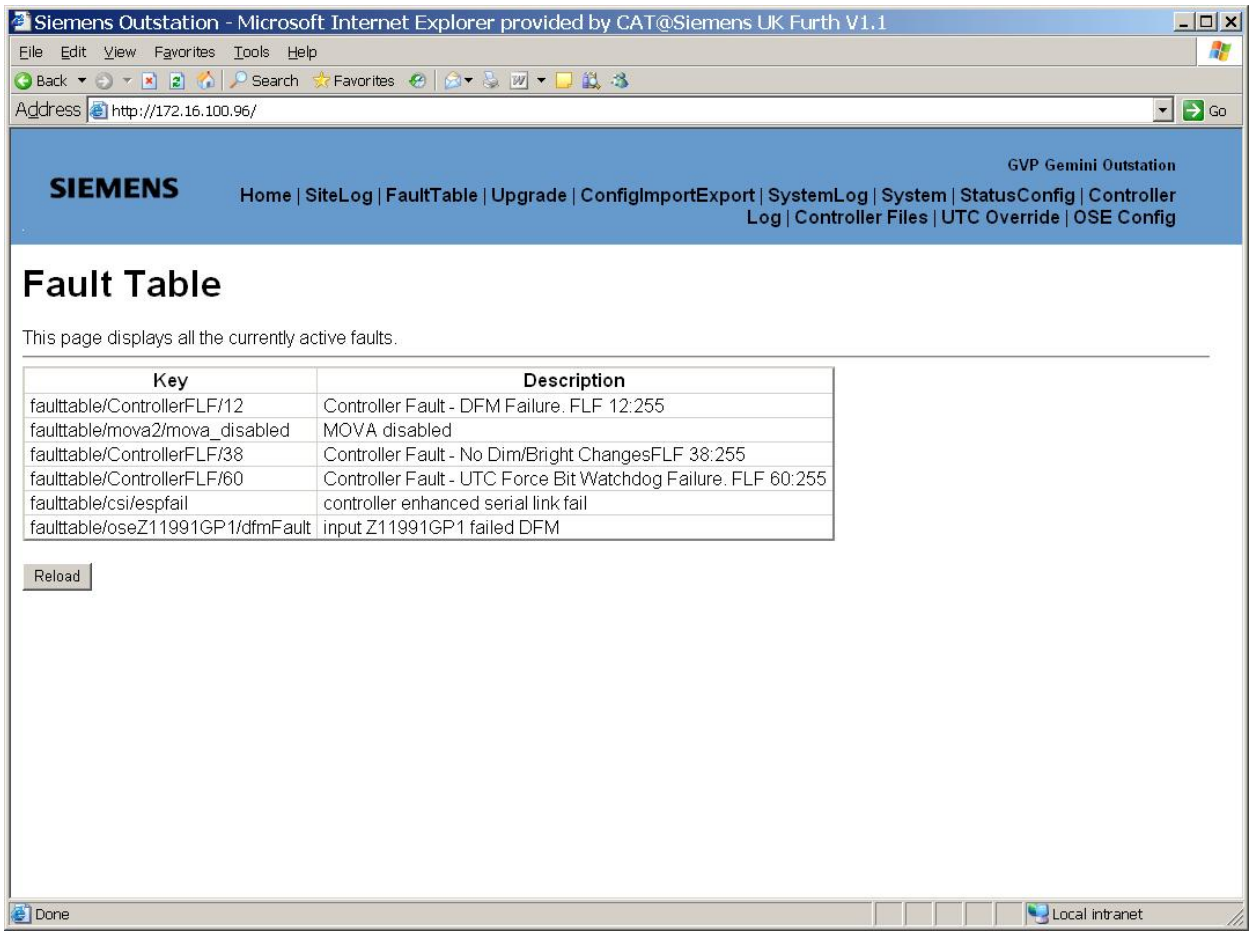
Pressing ‘Add to Site Log’ will create a new entry in the Site Log using the User and Text details already supplied. If a file has been selected, it will also be attached to the log entry.

4.2 FAULT TABLE SCREEN

This screen displays currently active faults on the outstation and on monitored street equipment. When the outstation is connected to an ST800/900 controller, the active controller faults are displayed (i.e. as provided by the FFS controller command).

Entries can be made in the table by each running application. The applications currently running are displayed on the ‘System’ screen.

Additional information on fault timing and fault history can be obtained by examination of the ‘System Log’ screen and, in the case of ST800/900 etc. controllers, the ‘Controller Log’ screen.

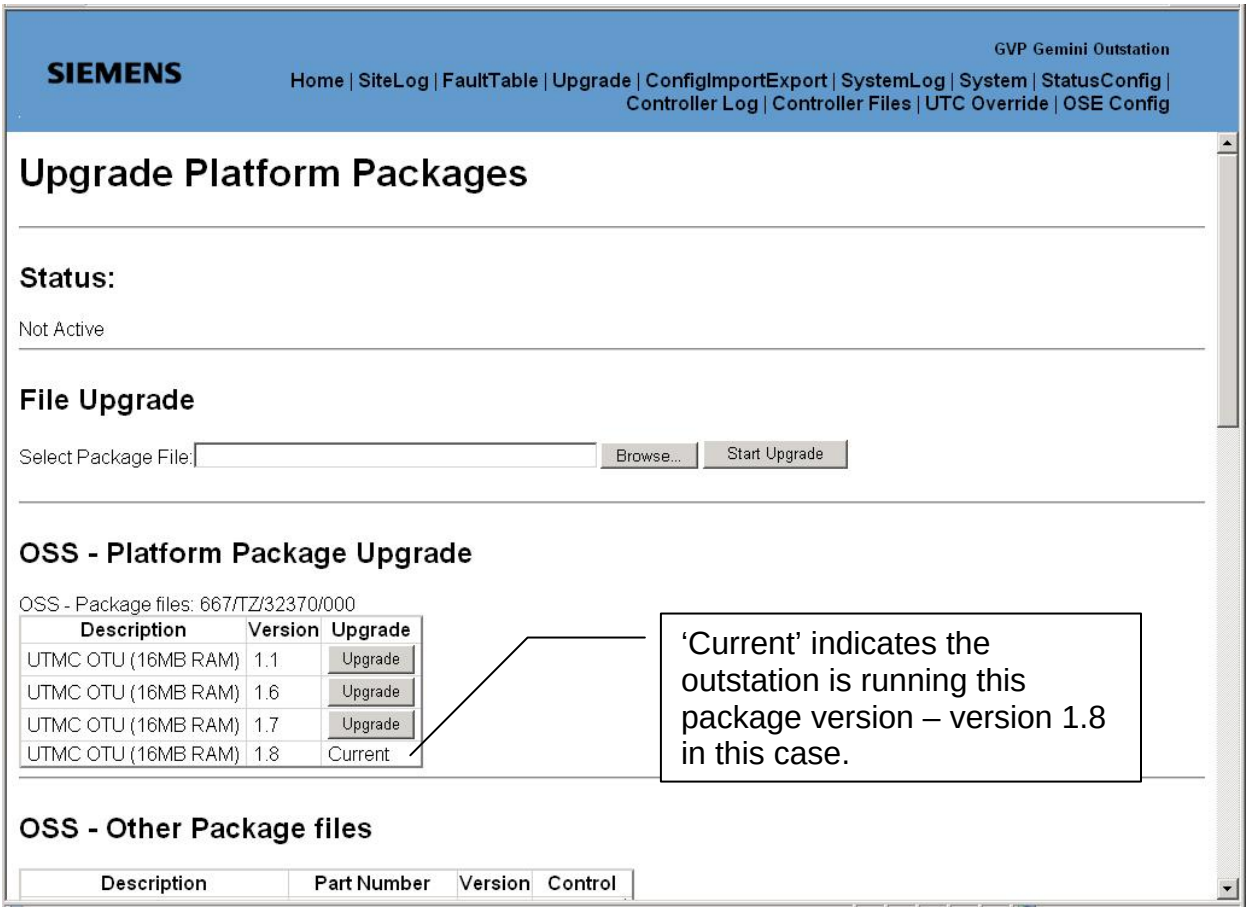


Reload

Pressing ‘Reload’ will update the fault table information.

4.3 UPGRADE SCREEN

This screen allows the user to load new versions of software (package files) onto the outstation.



Status:

Indicates if an upgrade is currently in progress. The sequence is:-

Starting TFTP download
Writing Data
Upgrade complete
Rebooting

File Upgrade

Use this interface where the new package file resides on the users PC.

Browse

Press 'Browse' and then navigate to the package file to be loaded.

Start Upgrade

Once the package file has been selected, press 'Start Upgrade' to start the process of transferring the new file to the outstation. The outstation will display a new screen when the transfer starts. Press the 'Back' button to view the progress of the upgrade. The outstation will reboot on completion of the transfer. Allow 1

minute for reboot and then select the ‘System’ screen to re-connect to the outstation and check the upgrade has completed.

OSS - Platform Package Upgrade

Use this interface where an Outstation Support Server (OSS) is configured. The table shows all the older and newer versions of the currently loaded package which are available at the OSS. The package version currently running on the outstation is identified by the ‘current’ designation.

Upgrade

Press ‘Upgrade’ to start the process of transferring the corresponding package version from the OSS to the outstation. The outstation will reboot on completion of the transfer. Allow 1 minute for reboot and then press ‘System’ to re-connect to the outstation and check the upgrade has completed.

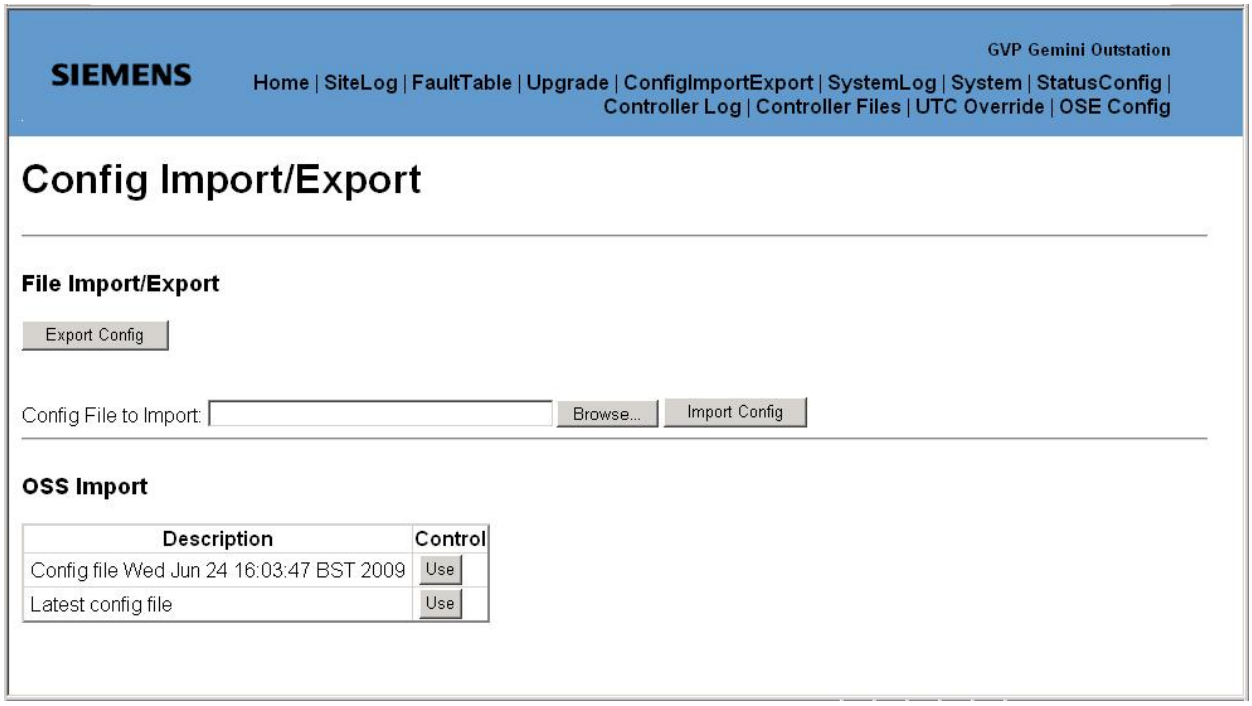
OSS - Other Package files

This table lists all the other package part numbers which are available at the OSS.

 : Switching to a different package part numbers should only be performed where it is known from handbook and other published information that the new package is compatible with the outstation hardware and that it provides the facilities required.

4.4 CONFIG IMPORT/EXPORT SCREEN

This screen allows the user to save and restore outstation configuration data. The configuration data is held in an XML formatted file which can be transferred via email, PC etc.



File Import/Export

Use this interface to save the current configuration file to the users PC or to restore a configuration from the users PC onto the outstation.

Export Config

Press ‘Export Config’ and then ‘Save’ the file in the desired location on the user PC. Files are named as

<Site Name>_<date>-<time>_config.xml

e.g. X12345_2009-09-18-15-18-11_ config.xml

Browse

Press 'Browse' and then navigate to the xml configuration file to be loaded.

Import Config

Press ‘Import Config’ to load the selected xml file from the users PC to the outstation.

> **:** The imported configuration file replaces all of the existing outstation configuration data and so should be used with caution. If, for example, the outstation IP address is different in the new configuration, then the outstation will start operating with the new IP address and be unavailable at the old address.

OSS Import

Use this interface where an Outstation Support Server (OSS) is configured. The table shows all automatically-saved versions of the outstation configuration which are available at the OSS. The outstation configuration is periodically compared with the latest available from the OSS and if the two are different then the new configuration is saved to the OSS. The table is time-ordered with the most recent version at the bottom of the table.

The backup of outstation configuration to the OSS can be enabled/disabled on the StatusConfig screen.

Use

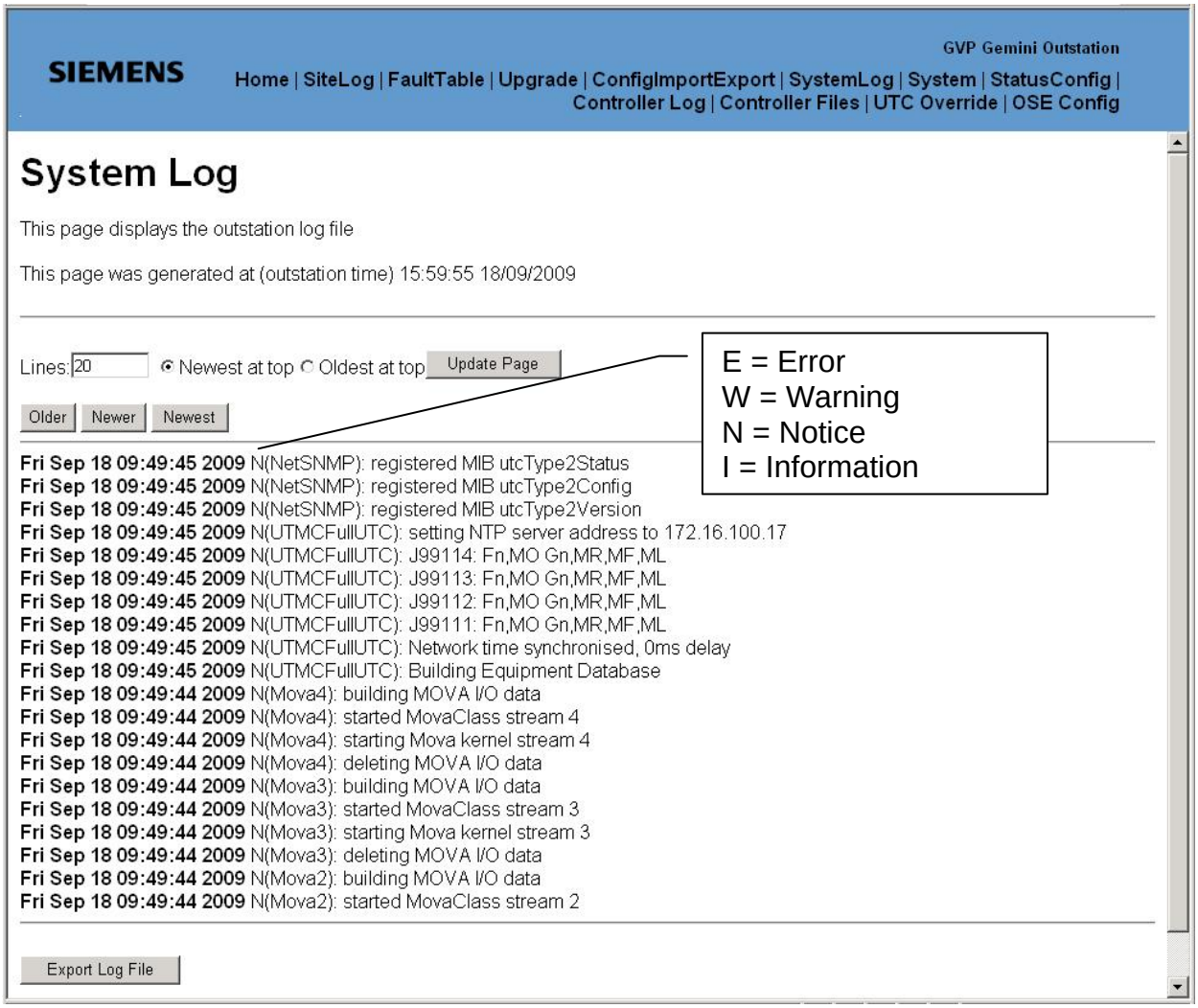
Press ‘Use’ to start the process of transferring the corresponding configuration data version from the OSS to the outstation.

> **:** The imported configuration file replaces all of the existing outstation configuration data and so should be used with caution. If, for example, the outstation IP address is different in the restored configuration, then the outstation will start operating with the restored IP address and be unavailable at the current address.

4.5 SYSTEM LOG SCREEN

This screen allows the user to display historic information logged by the software. The log is updated on the flash file system and maintained across power failures and package upgrades. When the log has grown to its maximum size, the oldest information in the log is discarded.

The amount of information collected in the log can be controlled using the “debug” command – see section 3.7.



Lines

This field sets the number of lines to be displayed on each page. Use 'Update Page' to apply the new setting.

Newest at top

Select this field to order the log entries so that the more recent events are at the top of the page. Use ‘Update Page’ to apply the new setting.

Oldest at top

Select this field to order the log entries so that the more recent events are at the bottom of the page and the older events are at the top of the page. Use 'Update Page' to apply the new setting.

Older

Change to the next oldest page of log entries.

Newer

Change to the more recent page of log entries.

Newest

Change to the most recent page of log entries.

Export Log File

Press ‘Export Log File’ and then ‘Save’ the file in the desired location on the user PC. Files are named as:-

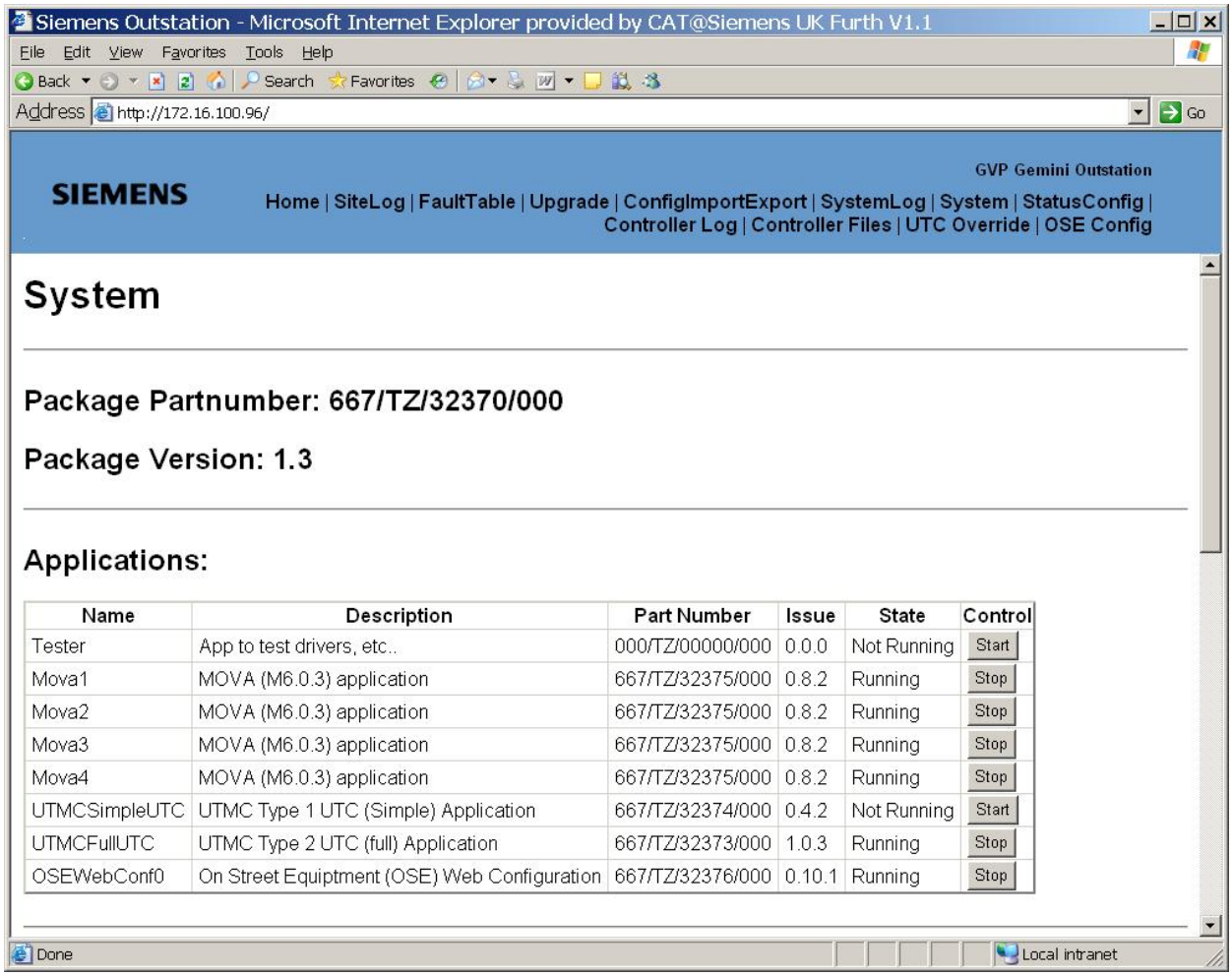
<Site Name>_<date>-<time>_log.wri

e.g. X12345_2009-09-18-15-18-11_ log.wri

4.6 SYSTEM SCREEN

This screen displays the list of software components running on the outstation. Applications can be started and stopped and the outstation can be re-booted.

4.6.1 System Screen – Part 1



Applications:

The table lists each software application available on the outstation, together with its status as either ‘running’ or ‘not running’. The Tester application is only used during production testing of some products.

Start

Press ‘Start’ to initiate the corresponding application.

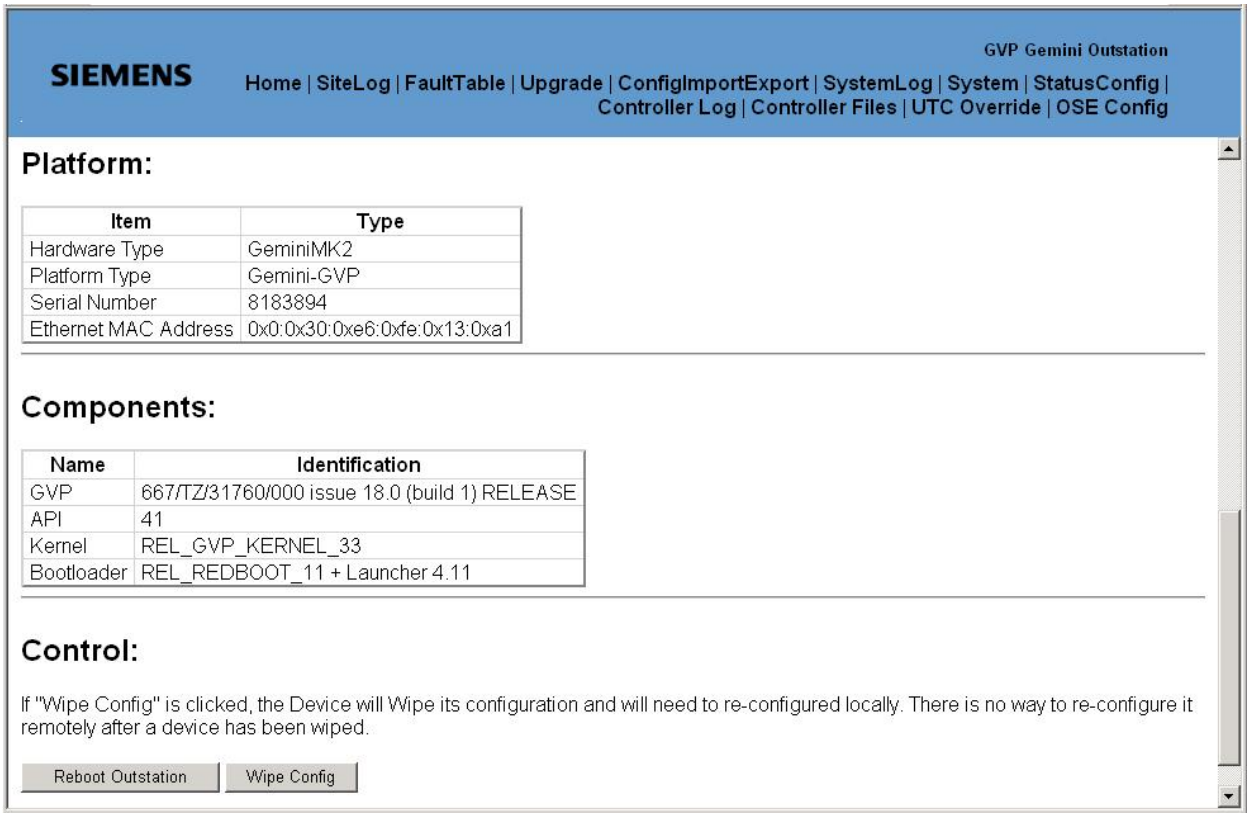
Stop

Press 'Stop' to close down the corresponding application.

Plugins:

The table lists the plugin components running on the outstation. Plugins are loaded where required by an application.

4.6.2 System Screen – Part 2



Platform:

This table gives information on the outstation hardware and related configuration.

Hardware Type

This entry identifies the CPU hardware type (currently only GeminiMK2 available)

Platform Type

This entry identifies the platform type (currently only Gemini-GVP available)

Serial Number

This is the serial number of the CPU card, also printed on the CPU card bar-code label.

Ethernet MAC Address

This is the unique Ethernet MAC Address assigned to the CPU card during production. It can be used to identify the unit on the network.

Component:

This table lists the various system software components, together with their version information.

GVP

Generic Versatile Platform part number and issue state.

API

This is the Application Programming Interface version number for GVP. All applications must be compatible with this version API of they will not be loaded by GVP.

Kernel

This is the release version of the eCos operating system.

Bootloader

This is the version of Redboot and the version of Launcher. Redboot manages the initial startup and then Launcher controls the loading of GVP from the flash file system.

Control:

Reboot Outstation

Press ‘Reboot Outstation’ to shut down all applications and initiate a restart of the outstation.

Wipe Config

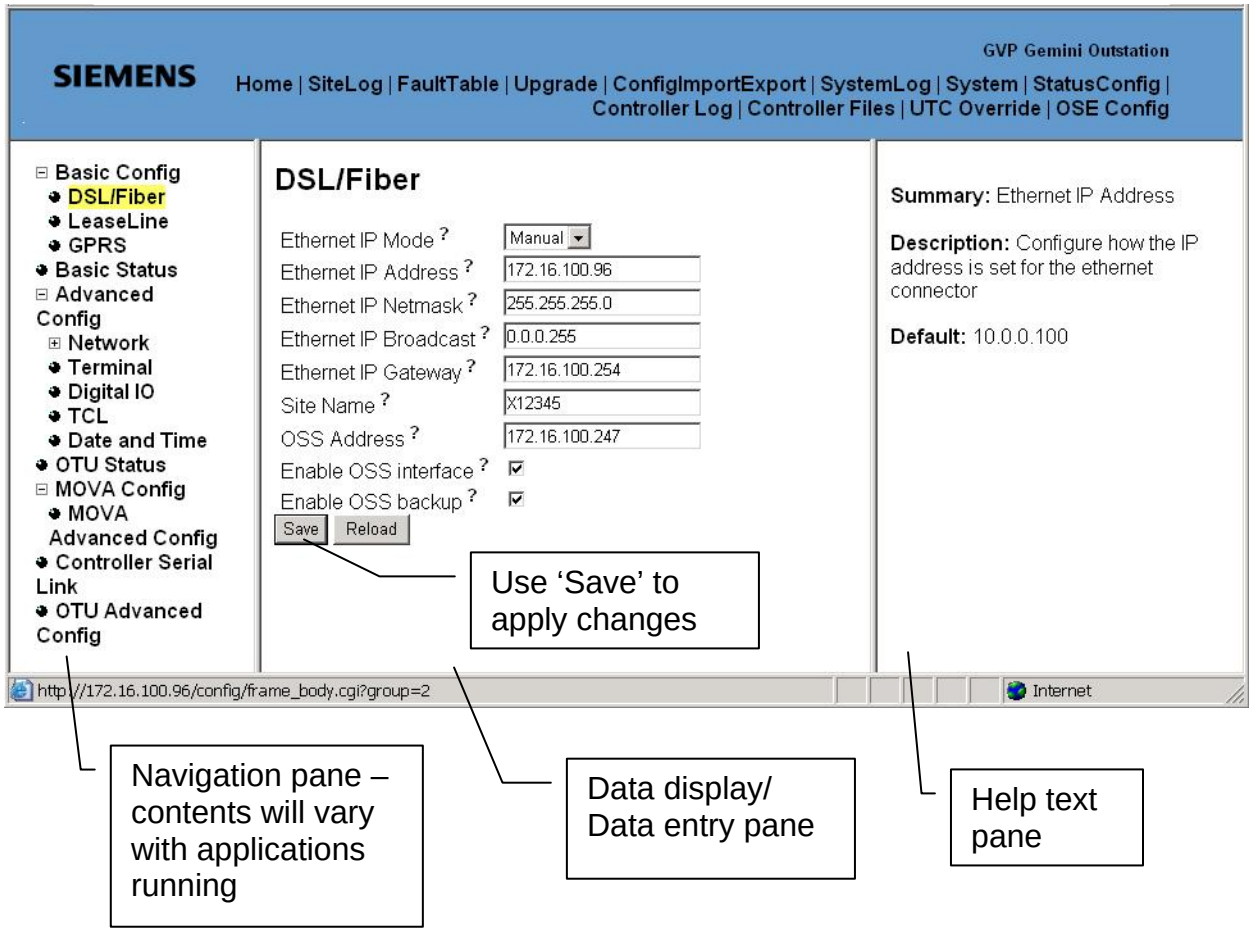
Press ‘Wipe Config’ to delete all outstation configuration data.

: If "Wipe Config" is clicked, the outstation will **ERASE** its configuration and will need to re-configured locally. There is no way to re-configure the outstation remotely after it has been wiped.

4.7 STATUS CONFIG SCREENS

This series of screens enables the user to display and edit the configuration settings of the outstation. After changing the value of a field, press 'Save' to save edits and apply changes to the running system.

4.7.1 DSL/Fiber Screen



?

Pressing the '**?**' next to a field will bring up help text in the right-hand window pane.

Ethernet IP Mode

Enable/disable the Ethernet by selecting from the list:-

Down – Ethernet is disabled

Manual – Ethernet is enabled using the address entered in the Ethernet IP Address field.

Ethernet IP Address

Set the IP address to be used on the Ethernet interface. The IP address is normally allocated by the Network Administrator.

Ethernet IP Netmask

Set the IP Netmask to be used on the Ethernet interface. The Netmask is normally allocated by the Network Administrator.

Ethernet IP Broadcast

Set the IP Broadcast Address to be used on the Ethernet interface.

Ethernet IP Gateway

Set the IP Gateway Address to be used on the Ethernet interface. The Gateway Address is normally allocated by the Network Administrator.

Site Name

Set the name used to identify the outstation. This would typically be the outstation SCN e.g. X12345.

OSS Address

Set the IP address of the Outstation Support Server, where this facility is available on the system. The OSS provides a central location for backup of outstation configuration data and management of outstation software package versions.

Enable OSS interface

Enable or Disable the Outstation Support Server interface.

Enable OSS backup

Enable or Disable the automatic backup of outstation configuration data to the Outstation Support Server interface.

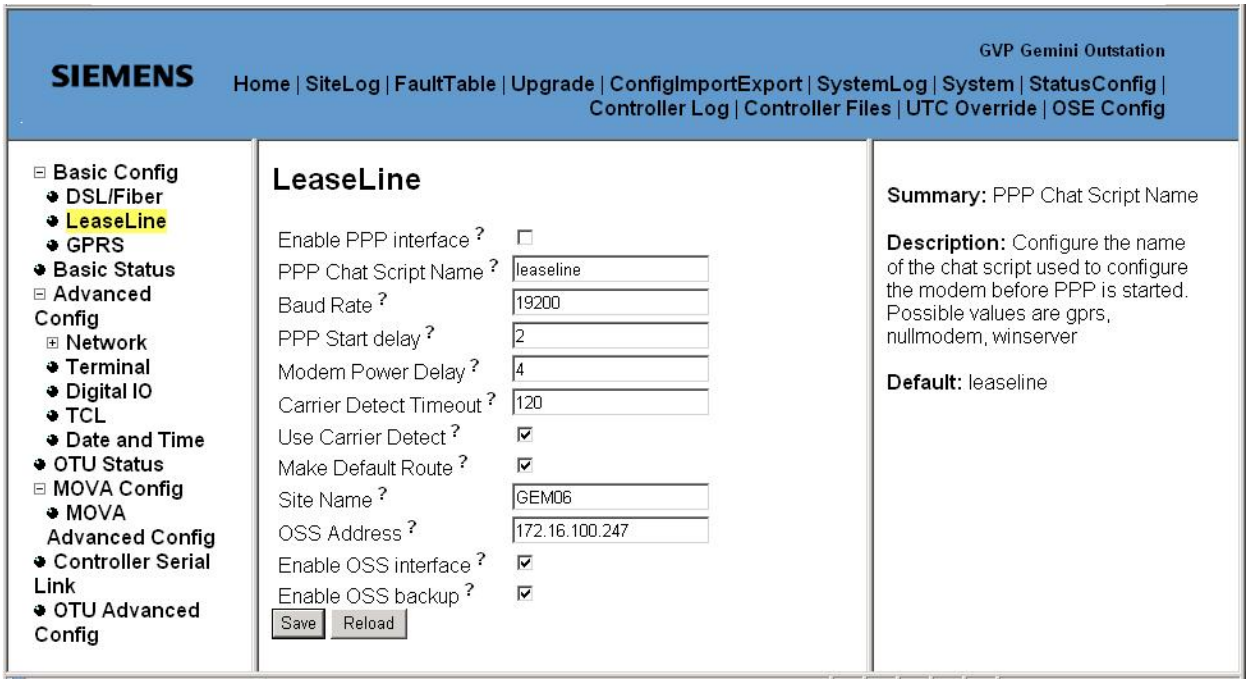
Save

Press ‘Save’ to save edits and apply changes to the running system.

Reload

Press ‘Reload’ to refresh the data on the screen.

4.7.2 Lease Line Screen



?

Pressing the '**?**' next to a field will bring up help text in the right-hand window pane.

Enable PPP interface

Enable or Disable the PPP (point-to-point) IP interface on the RS232 modem serial port.

PPP Chat Script Name

Configure the name of the chat script used to configure the modem before PPP is started. Possible values are leasline, gprs, nullmodem, winserver.

Baud Rate

This option defines the RS232 BAUD rate that will be used between the serial port and the Modem. Possible values are 9600, 19200, 38400, 57600, 115200

PPP Start delay

Configure the delay (in seconds) between modem power on and PPP starting. This is used to give the GPRS/other modem chance to connect upstream.

Modem Power Delay

Configure the delay (in seconds) to keep the modem power off.

Carrier Detect Timeout

Configure the delay (in seconds) to wait before giving up on Carrier Detect and restarting. This option is only used if the Use Carrier Detect option is enabled.

Use Carrier Detect

With this option, pppd will wait for the carrier detect to be active before starting PPP.

Make Default Route

With this option, pppd will make the PPP interface the default IP route for IP traffic.

Site Name

Set the name used to identify the outstation. This would typically be the outstation SCN e.g. X12345.

OSS Address

Set the IP address of the Outstation Support Server, where this facility is available on the system. The OSS provides a central location for backup of outstation configuration data and management of outstation software package versions.

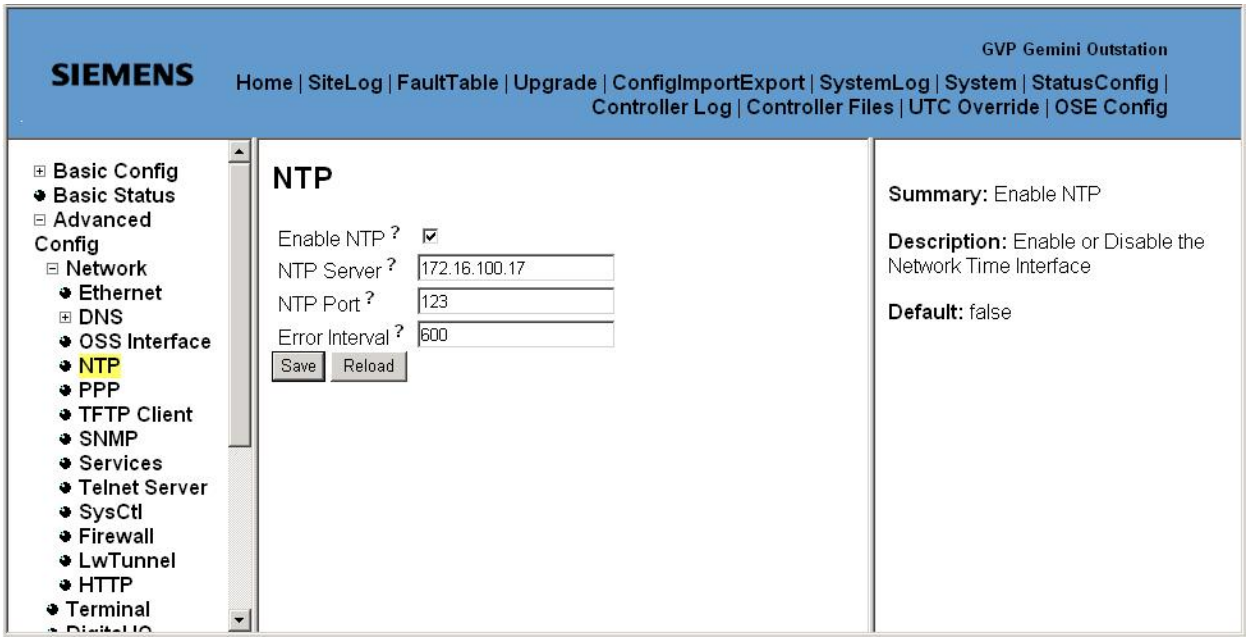
Enable OSS interface

Enable or Disable the Outstation Support Server interface.

Enable OSS backup

Enable or Disable the automatic backup of outstation configuration data to the Outstation Support Server interface.

4.7.3 NTP Screen



?

Pressing the '**?**' next to a field will bring up help text in the right-hand window pane.

Enable NTP

Enable or Disable the Network Time Interface.

NTP Server

Configure the IP address for the Network Time Server. This configuration item can be set automatically by some applications e.g. UTMC UTC type 2.

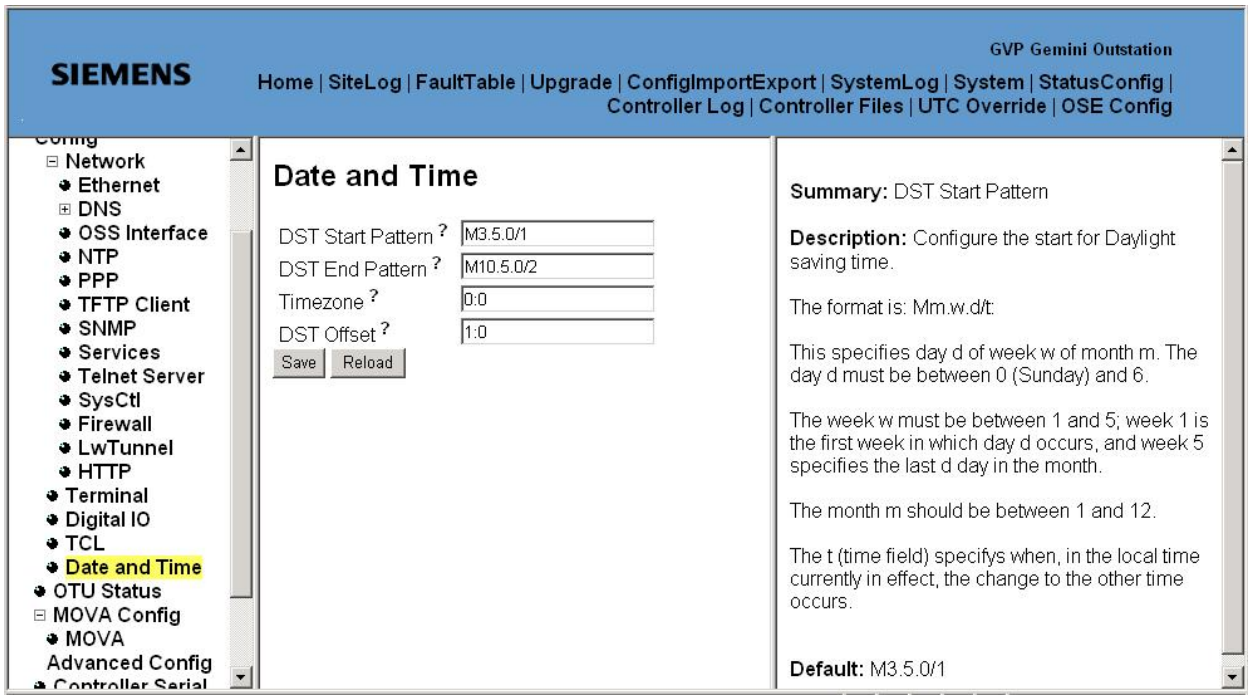
NTP Port

Configure the Port number for the Network Time Server.

Error Interval

Configure the number of seconds to wait before retrying after an error.

4.7.4 Date and Time Screen



?

Pressing the '**?**' next to a field will bring up help text in the right-hand window pane.

DST Start Pattern

Configure when Daylight saving will be introduced. The outstation local clock will be advanced by the DST offset (normally 1 hour).

The format is: Mm.w.d/t:
This specifies day d of week w of month m. The day d must be between 0 (Sunday) and 6.
The week w must be between 1 and 5; week 1 is the first week in which day d occurs, and week 5 specifies the last d day in the month.
The month m should be between 1 and 12.
The t (time field) specifies when, in the local time currently in effect, the change to the other time occurs.

DST End Pattern

Configure when Daylight saving will be cancelled.

The format is: Mm.w.d/t:
This specifies day d of week w of month m. The day d must be between 0 (Sunday) and 6.
The week w must be between 1 and 5; week 1 is the first week in which day d occurs, and week 5 specifies the last d day in the month.
The month m should be between 1 and 12.
The t (time field) specifies when, in the local time currently in effect, the change to the other time occurs.

Timezone

Configure the offset of the outstation local clock from GMT.

The format is: h:m:
Where h is hours and m is minutes.

DST Offset

Configure the additional offset the outstation should apply to the local clock during Daylight Saving Time.

The format is: h:m:
Where h is hours and m is minutes.

Appendix A – PC setup for Direct IP Connection

This appendix details how to connect a Gemini-2 running GVP to a Windows XP computer using the PC COM port and the Gemini-2 front handset connector.

For connection via Bluetooth, see Appendix B.

The following equipment is needed:

1. A PC with a serial port (USB or normal) running Windows XP
2. A 9-way (male) to 25-way (female) handset serial cable, see section A.4.

If the PC has not yet been configured using the GeminiConnect tool then follow instruction in section **Error! Reference source not found.**.

If the PC has already been configured using GeminiConnect then proceed as follows:-

1. Make sure the System LED on the Gemini unit is flashing.
2. Connect the Handset serial Cable between the Front handset connector on the outstation and the Serial Port on the Laptop/Desktop. If using Bluetooth, connect the Bluetooth dongle to the Front handset connector on the outstation.
3. Click on the desktop shortcut created by GeminiConnect e.g. :-

   Alternatively, from the Windows Start menu select **Start→Control Panel→Network Connections→GeminiConnect on COM...**
4. A Connection Windows will appear, and should connect to the outstation within 20 seconds.
5. To access the web interface click on the desktop shortcut created by GeminiConnect e.g. :-

   Alternatively, open internet explorer and use http://169.254.33.31/
6. To access the telnet interface, from the Windows Start menu select **Start->Run** then type the command **telnet 169.254.33.31**

A.1 Configuring the Direct Connection Using GeminiConnect Tool

This section describes how configure the Direct IP connection using the GeminiConnect software package.

1. Unzip the GeminiConnect package, available from the Siemens web site.
2. Double click on the **GeminiConnect.exe** file
3. Follow the on-screen instructions.

If the GeminiConnect software is not available, the configuration can be performed as described in the following sections. However, use of GeminiConnect is strongly advised, to make the installation as simple and automatic as possible.

A.2 Installing the PC Direct Connection Driver

This section describes how to manually configure the PC serial port to be a "Direct Communications Cable".

1. If the serial port is a non-standard one (Bluetooth/USB), make sure the Serial port is connected and working.
2. Open **Phone and Modem Options** from the Control-Panel
3. Click on the **Modems** tab at the top of the window.
4. Click **Add**
5. Select the **Don't detect my modem** option, click **Next**
6. From the left hand pane, select **(Standard Modem Types)**
7. From the right hand pane, select **Communications cable between two computers**
8. Click **Next**
9. Select **All Ports**, click **Next**
10. Click **Finish**

Once the driver is installed, the communications settings can be configured as described in the following section.

A.3 Configuring the Direct Connection Settings Manually

This section describes how configure the Direct IP connection using the standard Windows DUN (Dial-Up Networking) system. The steps only need to be performed once.

1. Make sure the Handset Serial Cable is DISCONNECTED
2. Open the **Network Connections** Window from the Control-Panel
3. Click **New Connection Wizard**, Click **Next**

4. Select **Set up an advanced connection**, Click **Next**
5. Select **Connect directly to another computer", Click *Next**
6. Select **Guest**, Click **Next**
7. Write the Text **GeminiConnection** for the **Computer Name**, Click **Next**
8. Select the correct Serial Communications port from the drop down list, Click **Next**
9. Select **Anyone's use**, Click **Next**
10. Click **Add a shortcut to this connection to my desktop**, Click **Finish**
11. For the username and password contact Siemens support.
12. Click **Save this user name and password for the following users**.
13. Select **Anyone who uses this computer**.
14. Click **Properties**
15. Select **General** tab. (should also be selected)
16. Under the **Select a device**, click **Configure**
17. Change the **Maximum speed (bps)** to 115200
18. Un-click all **Hardware features**, Click **OK**
19. Select **Options** tab.
20. Un-Click **Prompt for name and password, certificate, etc.**
21. Select **Networking** tab.
22. Select **Internet Protocol (TCP/IP)**
23. Click **Properties**
24. Click **Advanced**
25. Un-Click **Use default gateway on remote network**, Click **OK** on the Advanced TCP/IP settings window.
26. Click **OK** on the **Internet Protocol (TCP/IP) Properties** Window
27. Click **OK** on the **GeminiConnection Properties** Windows.
28. The Connection will start to dial, Just click **Cancel**

A.4 Handset Serial Cable Pinout

The wiring connections for a 9-way (male) to 25-way (female) handset serial cable is shown below. This cable is used when connecting to the outstation handset port from a PC.

DB9 (male)	Signal	DB25 (female)
1	-	-
2	RD	3
3	TD	2
4	DTR	20
5	SGND	7
6	DSR	6
7	RTS	4
8	CTS	5
9	-	-

Appendix B – PC setup for Bluetooth IP Connection

This appendix details how to connect a GeminiMK2 running GVP to a Windows XP computer via Bluetooth.

The following equipment is needed:

1. PC running Windows XP, with Bluetooth (integral or USB)
2. AIRserial3X serial Bluetooth adaptor 652/4/11360/000

B.1 Setting Up the Airserial3x Serial – Bluetooth Adaptor

1. Ensure that the DIP switches on the front of the adaptor are correctly set – the switches are numbered 1 to 4 and should be set UP, UP, UP, DOWN in numerical order.
2. Plug the adaptor into the front of the Gemini using the 25-9 way cable; a green power light should come on at the top of the unit indicating there is power.
3. The adaptor is now ready for pairing with another device.

B.2 Setting up the PPP over Bluetooth-serial connection

1. Start the new connection wizard [**Start → Settings → Network Connections → New connection wizard**]
2. Select "**Set up an advanced connection**" on the first screen and click "**Next**"
3. Select "**Connect directly to another computer**" and click "**Next**"
4. Select "**Guest**" and click "**Next**"
5. Name the connection "**Bluetooth link to Gemini**" and click "**Next**"
6. If you know the COM port of the serial-Bluetooth connection, pick it from the list and click "**Next**" otherwise – just click "**Next**"
7. Click through to finish the wizard until you are prompted to provide a Username and Password
8. Click the "**Properties**" button
9. Ensure the device selected is the COM port which represents the serial-Bluetooth link.
10. Click "**Configure**"
11. In the new window ensure the **Maximum speed (bps)** is set to **115200** and that all three hardware features (flow control, error control & modem compression) are unchecked. Click "**OK**" on all the open configuration windows.
12. Enter the username and password for the connection and click connect to test the connection

B.3 Setting up the Windows Bluetooth Link (IVT Blue Soliel / Widcomm)

1. Plug in the USB Bluetooth dongle and start the IVT BlueSoleil software from the start menu [**Start → Programs → IVT BlueSoleil → BlueSoleil**]
2. Perform a device discovery [**Main menu → My Bluetooth → Bluetooth Device Discovery**]
3. After a short pause icons should appear around the orange circle in the middle of the screen. If no icons appear – and the status bar at the bottom of the window reports "**No new devices found**" then an error has occurred.
4. There may be several icons in the centre of the blue screen; most likely mobile phones and/or PDAs. Look for an icon that looks like a computer, right click on it and select "**Pair Device**"
5. When prompted to enter a paring code – enter the number "1234" in to the dialogue box and click OK; you will only have 30 seconds in which to do this. If the pairing fails or no prompt is given during the pairing and there is more then one computer like icon on the screen go back to step 4 and repeat on the other icon(s) as required.
6. A successful pairing is indicated by a small red tick next to the device icon in question – the name of the device may also change from an ID (xx:xx:xx:xx:xx:xx) to AirXL.
7. Once pairing with the adaptor is complete; right click its icon and select "**Refresh Services**".
8. Wait for the status bar at the bottom of the window to read "**Service Discovery finished**" and then right click on the adaptor icon again to start the bluetooth serial service [Right click device icon → **Connect → Bluetooth serial**]
9. If presented with a dialogue box, click "Yes" and make a note of the new COM port. If no dialogue box is presented; the COM port information will be displayed in a notification popup in the bottom right of the screen.
10. A successfully established connection is indicated in the BlueSoleil window by a dotted line with a red circle running along it connecting the AirXL icon and the orange circle in the middle of the screen; a signal strength indicator may be present as well.
11. If the connection "**Bluetooth link to Gemini**" has already been set up then go to B.2 "Setting up the PPP over Bluetooth-serial connection" Step 9; otherwise go to B.2 "Setting up the PPP over Bluetooth-serial connection" Step 1.

B.4 Setting up the Windows Bluetooth Link (Advent Netbook)

1. Ensure that the Bluetooth radio is activated (indicated by blue indicator LED on the bottom right hand side of the laptop). If it is not; press Fn – F11 until the Bluetooth radio is online.
2. Open the Bluetooth settings manager [**Start → Bluetooth Settings**]

3. Click the “**New Connection**” button and select “**Custom mode**” from the options in the wizard, click “**Next**”
4. Select **AIRserial3X** from the list of discovered devices and click “**Next**”. If the device is not shown – ensure the adaptor is on and press refresh.
5. When prompted select the **Serial Port Service** (SPP) from the list of device services and click “**Next**”
6. Make a note of the COM port assigned to the Bluetooth link, click “**Next**” through the remaining screens to finish the wizard
7. Right click the AIRserial3X connection icon in the Bluetooth settings window and select the “**Connect**” option. Enter the PIN code if prompted.
8. Upon a successful connection – the icon should change, indicating a successful link.
9. If the connection “**Bluetooth link to Gemini**” has already been set up then go to “Setting up the PPP over Bluetooth-serial connection” Step 9; otherwise go to “Setting up the PPP over Bluetooth-serial connection” Step 1.

Appendix C

The text below is the licence for the GNU open software that is embedded in the Outstation firmware. This firmware contains modified GPL code and that source code is available on application to Siemens as per clause 3a or 3b.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

* a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

* b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

* c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

* a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

* b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

* c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by

court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

As a special exception, if other files instantiate templates or use macros or inline functions from this file, or you compile this file and link it with other works to produce a work based on this file, this file does not by itself cause the resulting work to be covered by the GNU General Public License. However the source code for this file must still be made available in accordance with section (3) of the GNU General Public License.

This exception does not invalidate any other reasons why a work based on this file might be covered by the GNU General Public License.

END OF TERMS AND CONDITIONS

C.1 INDEX

M

N

O

P

S

T

W

LAST PAGE OF THE

HANDBOOK