EndRun
TECHNOLOGIES
"Smarter Timing Solutions"

Tycho *GPS*

Frequency Reference

User Manual

Tycho *GPS*

Frequency Reference User Manual

Preface

Thank you for purchasing the Tycho Frequency Reference. Our goal in developing this product is to bring you a precise time and frequency reference that will quickly, easily and reliably meet or exceed your system requirements. Your new Tycho is fabricated using the highest quality materials and manufacturing processes available today, and will give you years of troublefree service.

About EndRun Technologies

EndRun Technologies is dedicated to the development and refinement of the technologies required to fulfill the demanding needs of the time and frequency community.

The instruments produced by EndRun Technologies have been selected as the timing reference for a variety of industries and applications - computer networks, satellite earth stations, power utilities, test ranges, broadcast and telecommunications systems and more.

EndRun Technologies is committed to fulfilling your precision timing needs by providing the most advanced, reliable and cost-effective time and frequency equipment available in the market today.

Trademark Acknowledgements

IBM-PC, Linux, NotePad, Timeserv, UNIX, Windows NT, WordStar are registered trademarks of the respective holders.

Part No. USM3021-0000-000 Revision 10
December 2007

About This Manual

This manual will guide you through simple installation and set up procedures.

Introduction – The Tycho, how it works, where to use it, its main features.
Basic Installation – How to connect, configure and test your Tycho.
Console Port – Description of the Linux console commands for use over the network and serial ports.

If you detect any inaccuracies or omissions, please inform us. EndRun Technologies cannot be held responsible for any technical or typographical errors and reserves the right to make changes to the product and manuals without prior notice.

Warranty

This product, manufactured by EndRun Technologies, is warranted against defects in material and workmanship for a period of two years from date of shipment, under normal use and service. During the warranty period, EndRun Technologies will repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to EndRun Technologies. Buyer shall prepay shipping charges to EndRun Technologies and EndRun Technologies shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to EndRun Technologies from another country.

Products not manufactured by EndRun Technologies but included as an integral part of a system (e.g. peripherals, options) are warranted for ninety days, or longer as provided by the original equipment manufacturer, from date of shipment.

Limitation of Warranty

The foregoing express warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer or User, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

TO THE EXTENT PERMITTED BY LAW, THIS WARRANTY AND REMEMDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, REMEDIES AND CONDITIONS WHETHER ORAL OR WRITTEN, STATUTORY, EXPRESS, OR IMPLIED. AS PERMITTED BY APPLICABLE LAW, ENDRUN SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Warranty Repair

If you believe your equipment is in need of repair, call EndRun Technologies and ask for a customer service agent. It is important to contact us first as many problems may be resolved with a phone call. Please have the serial number of the unit and the nature of the problem available before you call. If it is determined that your equipment will require service, we will issue an RMA number. You will be asked for contact information, including your name, address, phone number and e-mail address.

Ship the unit prepaid in the original container or a container of sufficient strength and protection to EndRun Technologies. EndRun will not be responsible for damage incurred during shipping to us. Be sure the RMA number is clearly identified on the shipping container. Our policy is to fix or repair the unit within 5 business days. If it is necessary to order parts or if other circumstances arise that require more than 5 days, an EndRun service technician will contact you.

Repair After Warranty Expiration

If the warranty period has expired, we offer repair services for equipment you have purchased from EndRun. Call and ask for a customer service agent. It is important to contact us first as many problems may be resolved with a phone call. Please have the serial number of the unit and the nature of the problem available before you call. If it is determined that the equipment has failed and you want EndRun to perform the repairs, we will issue you an RMA number. Ship the unit prepaid in the original container or a container of sufficient strength and protection to EndRun Technologies. EndRun will not be responsible for damage incurred during shipping to us. Customer is responsible for shipping costs to and from EndRun Technologies. Be sure the RMA number is clearly identified on the shipping container. After the equipment has been received we will evaluate the nature of the problem and contact you with the cost to repair (parts and labor) and an estimate of the time necessary to complete the work.

Limitation of Liability

The remedies provided herein are Buyer's sole and exclusive remedies. EndRun Technologies shall not be liable for any direct, indirect, special, incidental or consequential damages, whether based on contract, tort or any other legal theory.

EndRun Contact Information

Address: EndRun Technologies
1360 N. Dutton Avenue #200
Santa Rosa, CA 95401
Phone: (707)573-8633
Fax: (707)573-8619
Sales: 1-877-749-3878 or (707)573-8633
sales@endruntechnologies.com
Support: 1-877-749-3878 or (707)573-8633
support@endruntechnologies.com

Table of Contents

Chapter*One*

Introduction

The Tycho GPS Frequency Reference is a high-performance, cost-effective system that provides highly-precise time and frequency outputs. The Tycho includes 1 PPS and IRIG-B as standard outputs plus an RS-232 serial port. In addiition, a network port which includes many protocols including TELNET, FTP, DHCP, SNMP, and SSH is a standard Tycho feature.

The Tycho utilizes the GPS transmissions to precisely synchronize itself to Universal Coordinated Time (UTC) to the 100-nanoseconds level of accuracy. The frequency of the internal oscillator is disciplined to match the frequency of the UTC timescale to the low parts in 10^{13} level of accuracy over 24-hour observation intervals. The time and frequency outputs are coherent after initial GPS synchronization, and synchronization is maintained via 20-bit DAC frequency control, rather than phase stepping, to provide excellent short-term stability.

For more detailed information that is not included in this manual, and links to other sites, please visit our website: http://www.endruntechnologies.com. *There you can also download firmware upgrades, the latest manuals and other documentation.*

Main Features

Overview

The Tycho is composed of a Global Positioning System (GPS) time and frequency engine integrated with an IBM-PC compatible fanless, convection-cooled 133 MHz CPU with integral ethernet interface, an RS-232 serial port, and a power supply. Non-volatile storage of the embedded Linux operating system and the Tycho application software is via FLASH memory.

Standard Features

In addition to sourcing a precision 1PPS timing reference and an IRIG-B timecode output, your Tycho GPS Frequency Reference includes a network port. The Tycho can be managed via the network port or a local console on the RS-232 serial port. See ***Chapter 3 - Control and Status Commands*** for more information.

Secure Network Interface

An ethernet port is provided as a standard feature of the Tycho Frequency Reference with a wide variety of protocols including SNMP with Enterprise MIB, SSH, TELNET, and FTP. Refer to ***Chapter 2 - Basic Installation*** for information to help you set up your network interface. The inclusion of SNMP v3 and SSH provides a very secure network interface and allows you to safely perform monitoring and maintenance activities over the network. Security-conscious users can also disable any or all of the risky protocols such as Telnet, Time and Daytime. In addition, access via SSH, SNMP and Telnet can be restricted to specific hosts. Refer to ***Appendix C - SNMP*** and ***Appendix D - Security*** for further information.

Free FLASH Upgrades

Firmware and configurable hardware parameters are stored in non-volatile FLASH memory, so the Tycho can be easily upgraded in the field using FTP and TELNET or the local RS-232 serial I/O port. Secure upgrades are possible via SSH and SCP. We make all firmware upgrades to our products available to our customers free of charge. For firmware upgrade procedures refer to ***Appendix B - Upgrading the Firmware***.

GPS Timing-How It Works

The time and frequency engine in the Tycho receives transmissions from satellites that are operating in compliance with the Navstar GPS Interface Control Document (ICD) known as GPS-ICD-200. It specifies the receiver interface needed to receive and demodulate the navigation and time transfer data contained in the GPS satellite transmissions. The GPS navigation system requires a means of synchronizing the satellite transmissions throughout the constellation so that accurate receiver-to-satellite range measurements can be performed via time-of-arrival measurements made at the receiver. For the purposes of locating the receiver, measurements of the times-of-arrival of transmissions from at least four satellites are needed. For accurate time transfer to a receiver at a known position, reception of the transmissions from a single satellite is sufficient.

The GPS system designers defined *system time* to be *GPS time.* GPS time is maintained by an ensemble of high-performance cesium beam atomic frequency standards located on the earth's surface. GPS time is measured relative to UTC, as maintained by the United States Naval Observatory (USNO), and maintained synchronous with UTC-USNO except that it does not suffer from the periodic insertion of leap seconds. Such discontinuities would unnecessarily complicate the system's navigation mission. Contained in the data transmitted from each satellite is the current offset between GPS time and UTC-USNO. This offset is composed of the current integer number of leap seconds difference and a small residual error that is typically less than +/- 10 nanoseconds.

Each satellite in the constellation contains redundant cesium beam or rubidium vapor atomic frequency standards. These provide the timebase for all transmissions from each satellite. These transmissions are monitored from ground stations located around the world and carefully measured relative to GPS time. The results of these measurements for each satellite are then uploaded to that satellite so that they may be incorporated into the data contained in its transmissions. The receiver can use this data to relate the time-of-arrival of the received transmissions from that satellite to GPS time.

All of this means that during normal operation, the source of the timing information being transmitted from each of the satellites is directly traceable to UTC. Due to the nature of the GPS spread spectrum Code Division Multiple Access (CDMA) modulation scheme, this timing information may be extracted by a well-designed receiver with a precision of a few nanoseconds. The GPS time and frequency engine in the Tycho does just that.

Where to Use It

Since signals from the GPS satellites are available at all locations on the globe, you may deploy the Tycho virtually anywhere. However, you must be able to install an antenna with good sky visibility, preferably on the rooftop. Once synchronized, the Tycho can maintain acceptable accuracy for about a day without GPS reception, by flywheeling on its standard temperature compensated crystal oscillator (TCXO). For improved holdover for those using the Tycho as a frequency standard an oscillator upgrade is available.

Chapter *Two*

Basic Installation

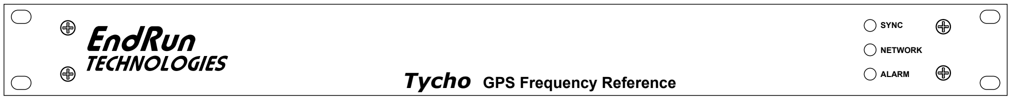
This chapter will guide you through the most basic checkout and physical installation of your Tycho GPS Frequency Reference. Subsequent chapters and appendices will give you the information needed to configure your installation for the maximum performance in your operating environment. Though some familiarity with Linux or other Unix-like operating systems would be helpful, it is not essential. When operating your Tycho with its standard network interface, basic familiarity with TCP/IP networking protocols like **`ping`**, **`telnet`** *and* **`ftp`** *is required.*

Checking and Identifying the Hardware

Unpack and check all the items using the shipment packing list. Contact the factory if anything is missing or damaged. The Tycho shipment typically contains:

- Tycho (part # 3021-0001-000 or #3019- variant)
- Tycho GPS User Manual (part #USM3021-0000-000)
- IEC 320 AC Power Cord (part #0501-0003-000)
  (This part will not be present if using the DC power option.)
- DB9F-to-DB9F Null Modem Serial I/O Cable (part #0501-0002-000)
- RJ-45 to RJ-45 CAT-5 patch cable, 2 meters (part #0501-0000-000)
- Antenna/cable assembly (part #0610-0006-001 or #0610- variant)

Tycho Physical Description



Sync LED

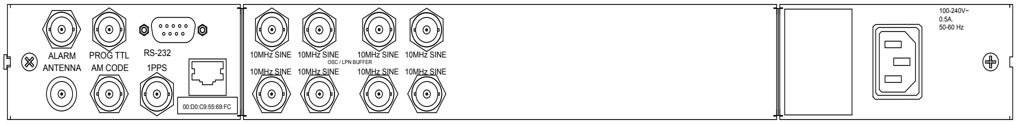
This green LED flashes to indicate synchronization status.

Network LED

This amber LED illuminates when the Tycho is connected to the network and flashes when receiving or transmitting packets..

Alarm LED

This red LED illuminates briefly at power-up, and thereafter whenever a serious fault condition exists.



Antenna Jack

This TNC connector mates with the downlead cable from the external antenna.

RS-232 Connector

This DB-9M connector provides the RS-232 serial I/O console interface to the Tycho. This console allows the user to initialize and maintain the Tycho. See ***Chapter 3 - RS-232 Serial I/O Port Signal Definitions*** for detailed information.

10/100Base-T Jack

This RJ-45 connector mates with the ethernet twisted pair cable from the network.

1PPS Jack

This BNC connector provides the 1PPS TTL output. The pulse width is normally 1 millisecond wide when shipped from the factory but can be changed via console command **`cpuoptsconfig`**. Other selections are 20 microseconds, 100 milliseconds and 500 milliseconds. See signal definition in ***Appendix H - Specifications*** for the 1PPS Output.

AM Code Jack

This BNC connector provides the amplitude-modulated timecode output. The timecode output is normally IRIG-B122 when shipped from the factory, but can be changed via the console command **`cpuoptsconfig`**. Other selections are available. See signal definition in ***Appendix H - Specifications*** for the AM Code Output.

Alarm Jack
(Option)

This BNC connector (or terminal strip) provides the optional alarm output, and is usually not installed. If installed, see description in ***Chapter 4 - Optional Rear-Panel Outputs*** and signal definition in ***Appendix H - Specifications.***

Prog TTL Jack
(Option)

This BNC connector provides the optional Programmable TTL pulse rate output and is usually not installed. If installed, see description in ***Chapter 4 - Optional Rear-Panel Outputs*** and signal definition ***in Appendix H - Specifications.***

10 MPPS or 100 PPS, etc.
(Option)

This BNC connector provides an optional customer-specified rate output and is usually not installed. If installed, it will be labeled for the appropriate rate such as “10 MPPS” or “100 PPS”, etc. This output is set at the factory and cannot be changed. See details in ***Chapter 4 - Optional Rear-Panel Outputs*** and signal definition ***in Appendix H - Specifications*** for the Fixed Rate Output.

1PPS (RS-422)
(Option)

This optional DB-9M connector provides the 1PPS output at RS-422 levels and is usually not installed.. The pulse width is normally 1 millisecond wide when shipped from the factory but can be changed via command **`cpuoptsconfig`**. See pinout details in ***Appendix H - Specifications*** for the 1PPS RS-422 Output.

Serial Time
(Option)

This optional DB-9M connector provides the serial I/O interface with a once-per-second ASCII time string output and is usually not installed. For further information refer see description in ***Chapter 4 - Optional Rear-Panel Outputs*** and ***Appendix H - Specifications.***

Output Boards
(Options)

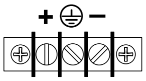
Your Tycho has two option slots that can be configured with a variety of output buffer modules. In Tycho products these output modules are not field-installable but must be installed at the factory. See ***Chapter 4 - Optional Rear-Panel Outputs*** for more information.

AC Power Input Jack

This IEC 320 standard three-prong connector provides AC power.

DC Power Input Block

This optional 3-position terminal block provides connection to the DC power source, and replaces the AC power input jack. See details in ***Appendix H - Specifications.***



Performing an Initial Site Survey

Using the status LED indicators, it’s easy to find out if your Tycho will work in your desired location:

1. Mount the antenna on the roof using the supplied mounting hardware. Make sure that it is not blocked by large metallic objects closer than one meter.

2. Screw the TNC plug on the end of the antenna cable onto the TNC antenna input jack on the chassis rear panel of the Tycho.

3. Plug one end of the supplied AC power cord into an 85-270 VAC outlet.

4. Plug the other end into the AC input connector on the chassis rear panel of the Tycho.

Initially upon power up:

1. The unit will light the red Alarm Status LED for about ten seconds.

2. Then it will continuously light the green Sync Status LED.

3. When the unit locks onto a GPS signal and begins to decode the timing data and adjust the local oscillator, the green Sync Status LED will flash very rapidly (about a 6 Hz rate) until the data is fully decoded and the local oscillator is fully locked to the GPS frequency. Note: If your unit has an oscillator upgrade (MS-OCXO or Rb) then it will need 5-10 minutes of warmup before trying to acquire a signal.

4. Then the green Sync Status LED will pulse at precisely a 1 Hz rate, synchronized to UTC seconds, with a short on duration relative to the off duration.

At this point, the GPS time and frequency engine has fully synchronized, and you may proceed to permanently mount the chassis and antenna in their desired locations. If you are unable to achieve GPS lock after 24 hours call Customer Support (1-877-749-3878) for assistance.

Installing the Tycho

FCC NOTICE

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmul interference in which case the user will be required to correct the interference at his own expense.

Mount the Tycho

Using standard 19” rack mounting hardware, mount the unit in the desired location. After mounting the unit and connecting the antenna cable, verify that it still acquires and tracks a GPS signal.

CAUTION

Ground the unit properly with the supplied power cord.

Position the power cord so that you can easily disconnect it from the Tycho.

Do not install the Tycho where the operating ambient temperature might exceed 122°F (50°C).

Connecting the DC Power Option

Connect the safety ground terminal to earth ground. Connect the “+” terminal to the positive output of the DC power source. Connect the "-" terminal to the negative output of the DC power source. Note that the Tycho has a "floating" internal power supply, therefore either the positive or negative output of the DC power source can be referenced to earth ground. This unit will not operate if the +/- connections are reversed; however it will not be damaged by a reverse connection.

SHOCK/ENERGY HAZARD

Install in Restricted Access Location.

Use 10-14 AWG copper wire only.

Terminal block screw torque: 9 in-lbs (1 nM).

Branch circuit must have circuit breaker, 15A or less.

Install terminal block cover after wiring.

Connecting and Configuring Ethernet

Connect one end of the CAT-5 patch cable supplied with your Tycho to the rear-panel-mounted RJ-45 connector labeled 10/100BASE-T. Connect the other end of the patch cable to your network through a 'straight' port on your hub. Do not connect it to a 'crossover' port on your hub.

By factory default, the Tycho will attempt to configure the ethernet interface automatically via the Dynamic Host Configuration Protocol (DHCP). The Tycho will attempt to set the netmask, its IP address, the IP address of the default gateway, the domain name and the IP addresses of any nameservers, if the DHCP server is configured to provide them. You may optionally configure the Tycho to also set its hostname via DHCP, if your DHCP server is configured to provide it. You can do this by running a simple shell script called **`netconfig`** after your unit is up on the network.

If your network *does* use DHCP for host configuration, and you are in a hurry to get your Tycho up and running, you may procede to ***Verifying Network Configuration*** to make sure that the network parameters were set up correctly. Otherwise, it is recommended that you read the following sections on use of the RS-232 serial I/O port now, since they will help you in debugging any problems that you may encounter with the automatic configuration via DHCP.

If your network *does not* use DHCP, you will need to configure your ethernet interface using the RS-232 serial I/O port. The following sections contain brief descriptions on how to do that.

Configuring Ethernet with the Serial Port

To configure your ethernet interface with the serial port, after logging in as the *root* user, you must run a simple shell script called **`netconfig`** from the **`bash`** shell prompt. This shell script will prompt you for the needed information and perform some syntax checking on your inputs. Then it will create or modify the appropriate files needed to configure the ethernet interface. The following sections will guide you in setting up communications with the Tycho using its RS-232 serial I/O port.

Connect the RS-232 Serial I/O Port

You will need to use the RS-232 serial I/O port if your network does not support the Dynamic Host Configuration Protocol (DHCP). In that case, you must be able to configure the Tycho network parameters manually using the Linux console shell interface which is provided by this serial I/O port. Under certain conditions, you may also need to use the RS-232 serial I/O port if you encounter a problem while upgrading the firmware in your Tycho.

To test serial communications with the Tycho you will need either a VT100 compatible terminal or a terminal emulation program running on your computer. We will refer to either of these as “terminal” for the remainder of this instruction.

1. Disconnect power from the Tycho.

2. Connect one end of the DB9F-to-DB9F null modem adapter cable to the serial I/O jack on the Tycho.

3. Connect the other end of the DB9F-to-DB9F null-modem adapter cable to the terminal. If the serial I/O port on your terminal does not have a DB9M connector, you may need to use an adapter. Refer to ***Chapter 3 - RS-232 Serial I/O Port Signal Definitions*** for details on the signal wiring. *If you are using a computer for your terminal, remember which port you are using because you will need to know that in order to set up your terminal software.*

Test the Serial Port

You must configure your terminal to use the serial I/O port you used in ***Connect the RS-232 Serial I/O Port***. You must also configure your terminal to use the correct baud rate, number of data bits, parity type and number of stop bits. *Be sure to turn off any hardware or software handshaking.* The settings for the Tycho are:

- 19200 is the Baud Rate
- 8 is the number of Data Bits
- None is the Parity
- 1 is the number of Stop Bits

After configuring these parameters in your terminal, apply power to the Tycho. After about 20 seconds, your terminal should display a sequence of boot messages similar to these:

```
***************************************************
* 6010-0040-000 Linux Bootloader v1.00 08/17/2004 *
***************************************************
Default root file system: FACTORY
To override and boot the UPGRADE partition type ‘UPGRADE’ within 5 seconds...
.....
```

These lines are the Linux bootloader boot prompt. This prompt will timeout after 5 seconds and the Linux kernel and the factory default Tycho root file system will be loaded. When the Linux kernel is loaded from FLASH memory into RAM a long list of kernel-generated, informational messages is displayed as the kernel begins execution and the various device drivers are initialized:

```
Booting Linux with FACTORY root file system...

6010-0041-000 Linux Kernel v2.4.26-1 #0 Wed Aug 18 17:28:45 UTC 2004
BIOS-provided physical RAM map:
BIOS-88: 0000000000000000 - 000000000009f000 (usable)
BIOS-88: 0000000000100000 - 0000000002000000 (usable)
32MB LOWMEM available.
On node 0 totalpages: 8192
zone(0): 4096 pages.
zone(1): 4096 pages.
zone(2): 0 pages.
DMI not present.
Kernel command line: config=11000001 initjffs=0 console=ttyS0,19200 root=/dev/
```

```
mtdblock4 load_ramdisk=1 rw
Initializing CPU#0
Calibrating delay loop... 66.96 BogoMIPS
Memory: 30784k/32768k available (812k kernel code, 1596k reserved, 162k data, 68k
init, 0k highmem)
Checking if this processor honours the WP bit even in supervisor mode... Ok.
Dentry cache hash table entries: 4096 (order: 3, 32768 bytes)
Inode cache hash table entries: 2048 (order: 2, 16384 bytes)
Mount cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 8192 (order: 3, 32768 bytes)
CPU: AMD 486 DX/4-WB stepping 04
Checking ‘hlt’ instruction... OK.
POSIX conformance testing by UNIFIX
PCI: Using configuration type 1
PCI: Probing PCI hardware
PCI: Probing PCI hardware (bus 00)
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Starting kswapd
JFFS2 version 2.1. (C) 2001 Red Hat, Inc., designed by Axis Communications AB.
Serial driver version 5.05c (2001-07-08) with MANY_PORTS SHARE_IRQ SERIAL_PCI enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
ttyS02 at 0x03e8 (irq = 0) is a ST16654
ttyS03 at 0x02e8 (irq = 3) is a ST16654
sc520_wdt: CBAR: 0x800df000
sc520_wdt: MMCR Aliasing enabled.
sc520_wdt: WDT driver for SC520 initialised.
RAMDISK driver initialized: 16 RAM disks of 16384K size 1024 blocksize
pcnet32.c:v1.28 02.20.2004 tsbogend@alpha.franken.de
PCI: Enabling device 00:0d.0 (0000 -> 0003)
pcnet32: PCnet/FAST III 79C973 at 0x1000, 00 0e fe 00 00 33
    tx_start_pt(0x0c00):~220 bytes, BCR18(9a61):BurstWrEn BurstRdEn NoUFlow
    SRAMSIZE=0x1700, SRAM_BND=0x0800, assigned IRQ 12.
eth0: registered as PCnet/FAST III 79C973
pcnet32: 1 cards_found.
Tempus SC520 flash device: 1000000 at 2000000
 Amd/Fujitsu Extended Query Table v1.3 at 0x0040
number of CFI chips: 1
Creating 7 MTD partitions on “Tempus SC520 Flash Bank”:
0x00000000-0x000e0000 : “Tempus kernel”
mtd: Giving out device 0 to Tempus kernel
0x000e0000-0x00100000 : “Tempus Lo BootLdr”
mtd: Giving out device 1 to Tempus Lo BootLdr
0x00100000-0x00200000 : “Tempus /boot”
mtd: Giving out device 2 to Tempus /boot
0x00200000-0x00300000 : “Tempus /logs”
mtd: Giving out device 3 to Tempus /logs
0x00300000-0x00900000 : “Tempus FACTORY rootfs”
mtd: Giving out device 4 to Tempus FACTORY rootfs
0x00900000-0x00fe0000 : “Tempus UPGRADE rootfs”
mtd: Giving out device 5 to Tempus UPGRADE rootfs
0x00fe0000-0x01000000 : “Tempus Hi BootLdr”
mtd: Giving out device 6 to Tempus Hi BootLdr
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 2048)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
mtdblock_open
ok
```

```
RAMDISK: Compressed image found at block 0
mtdblock_release
ok
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 68k freed
INIT: version 2.76 booting
/etc/rc.d/rc.S: /bin: is a directory
mtdblock_open
ok
mtdblock_open
ok
Loading GPS
Fri Aug 20 00:53:54 2004 -0.707128 seconds
2004
Setting system time using hwclock
INIT: Entering runlevel: 3
Entering multiuser...
Attempting to configure eth0 by contacting a DHCP server...
```

At this point, if you do not have a DHCP server configured on your network the unit will time-out and print these messages:

```
Tycho GPS DHCP Client was unable to find the DHCP Server!
Fix the problem and re-boot or set up static IP address
by running netconfig.
dnsdomainname: Host name lookup failure
(none)
```

Then these messages are printed, in either case:

```
Disabling IPv4 packet forwarding...
Starting daemons:  syslogd klogd inetd
Starting the System Time daemon...
Starting the SNMP daemon...
Starting the system logfile manager...
Starting the system watchdog...woof!
```

During this process, the factory default TychoGPS_0 root file system is loaded from FLASH disk to an 16MB ramdisk and the remainder of the boot process completes. At this point, the Tycho login prompt is displayed:

```
********************************************************************************
*           Welcome to Tycho GPS console on:  gsys.your.domain
*           Tue Feb 20  2001 21:47:03 UTC
********************************************************************************

gsys login:
```

Here you may log in as “gsysuser” with password “Praecis” or you may log in as the “root” user with password “endrun_1”. When logged in as “gsysuser”, you may check status information and view log files but you will not be able to modify any system settings or view secure files. In order to perform system setup procedures, which includes configuring the IP network settings, you must log in as the “root” user. After correctly entering the password at this prompt,

```
password:
```

the sign on message is shown. It identifies the host system as Tycho GPS and shows the software part number, version and build date:

```
Tycho GPS 6010-0042-000 v 1.00 Wed May  9 14:17:44 UTC 2002
Tycho GPS (root@gsys:~)->
```

This last line is the standard Tycho shell prompt. The Tycho uses the **`bash`** shell, which is the Linux standard, full-featured shell. After configuring the unit, you should change the passwords using the **`gsyspasswd`** command issued from the shell prompt.

If you do not see characters displayed by your terminal program within 30 seconds after the unit is powered up, you must troubleshoot your setup. An incorrectly wired cable or incorrect port setting in your terminal emulation program are the most common problems. Refer to ***Chapter 3 - RS-232 Serial I/O Port Signal Definitions*** for the signal connections for the Tycho.

NOTE

You must use a null-modem cable or adapter if you are connecting the Tycho to another computer or other equipment configured as Data Terminal Equipment (DTE). The supplied cable is a null-modem cable.

Once you have successfully established communications with the Tycho, you may procede to configuring the network parameters. Then you can communicate with the Tycho over the network using **`telnet`** or **`ssh`**.

Using netconfig to Set Up Your IP

The following is a sample transcript which illustrates the use of **`netconfig`**. The entries made by the user are underlined and are provided purely for illustrative purposes. You must provide equivalent entries that are specific to your network. Those shown here are appropriate for a typical network that does not use DHCP. Start the configuration process by typing **`netconfig`** at the shell prompt:

```
Tycho GPS(root@gsys)-> netconfig

********************************************************************************
******************** Tycho GPS Network Configuration  **************************
********************************************************************************
*                                                                              *
*   This script will configure the TCP/IP network parameters for your          *
*   Tycho GPS. You will be able to reconfigure your system at any time         *
*   by typing:                                                                 *
*                                                                              *
*   netconfig                                                                  *
*                                                                              *
*   The settings you make now will not take effect until you restart your      *
*   Tycho GPS, so if you make a mistake, just re-run this script before        *
*   re-booting.                                                                *
*                                                                              *
*   You will be prompted to enter your network parameters now.                 *
*                                                                              *
********************************************************************************
********************************************************************************

---DHCP Settings
Use a DHCP server to configure the ethernet interface? ([y]es, [n]o) n

---HOST name setting
```

```
Set the hostname of your Tycho GPS. Only the base
hostname is needed, not the domain.
Enter hostname: gsys

---DOMAIN name setting

Set the domain name. Do not supply a leading ‘.’
Enter domain name for gsys: your.domain

---STATIC IP ADDRESS setting

Set the IP address for the Tycho GPS. Example: 111.112.113.114
Enter IP address for gsys (aaa.bbb.ccc.ddd): 192.168.1.245

---DEFAULT GATEWAY ADDRESS setting

Set the default gateway address, such as 111.112.113.1
If you don’t have a gateway, just hit ENTER to continue.
Enter default gateway address (aaa.bbb.ccc.ddd): 192.168.1.241

---NETMASK setting

Set the netmask. This will look something like this: 255.255.255.0
Enter netmask (aaa.bbb.ccc.ddd): 255.255.255.248

Calculating the BROADCAST and NETWORK addresses...
Broadcast = 192.168.1.247     Network = 192.168.1.240

Your Tycho GPS’s current IP address, full hostname, and base hostname:
192.168.1.245       gsys.your.domain    gsys

---DOMAIN NAMESERVER(S) address setting

Will your Tycho GPS be accessing a nameserver ([y]es, [n]o)? y

Set the IP address of the primary name server to use for domain your.domain.
Enter primary name server IP address (aaa.bbb.ccc.ddd): 192.168.1.1

Will your Tycho GPS be accessing a secondary nameserver ([y]es, [n]o)? y

Set the IP address of the secondary name server to use for domain your.domain.
Enter secondary name server IP address (aaa.bbb.ccc.ddd): 192.168.1.2

Setting up TCP/IP...
Creating /etc/HOSTNAME...
Creating /etc/rc.d/rc.inet1...
Creating /etc/networks...
Creating /etc/hosts...
Creating /etc/resolv.conf...

********************************************************************************
********************************************************************************
*                                                                              *
*          The Tycho GPS network configuration has been updated.               *
*                                                                              *
*             Please re-boot now for the changes to take effect.               *
*                                                                              *
********************************************************************************
********************************************************************************
```

Verify Network Configuration

If you have made changes to your network configuration using `netconfig`, you should shutdown the Tycho and re-boot it. There are two ways to do this:

1. Cycle power to the Tycho.

2. Issue the shutdown with re-boot command at the shell prompt:

```
Tycho GPS(root@gsys:~)-> shutdown -r now
```

If you are using the RS-232 serial I/O port to communicate with the Tycho, you will be able to see the kernel generated boot messages when the unit re-boots. You should note the line

```
Configuring eth0 as 192.168.1.245...
```

if you have set up a static IP address, or this line

```
Attempting to configure eth0 by contacting a DHCP server...
```

if you are using DHCP. It appears near the end of the kernel generated boot messages.

If you are using DHCP and are not using the RS-232 serial I/O port, you will have to check the DHCP configuration information maintained by your DHCP server to determine the expected IP address and log in to the Tycho using `telnet` or `ssh` to verify successful DHCP configuration. Refer to the subsequent topics in this section ***Using Telnet*** and ***Using SSH***, for details on logging in to the Tycho that way. Once you have logged in, you may perform the following checks.

If you are not using DHCP, the IP address shown should match the static IP address which you entered during the `netconfig` procedure. If so, log in as “root” at the login prompt and check the other configuration parameters using `ifconfig`:

```
Tycho GPS(root@gsys:~)-> ifconfig

eth0      Link encap:Ethernet  HWaddr 00:0E:FE:00:00:34
          inet addr: 192.168.1.245 Bcast:192.168.1.247 Mask:255.255.255.248
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3779 errors:0 dropped:0 overruns:0 frame:0
          TX packets:727 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:5 Base address:0x300

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:170 errors:0 dropped:0 overruns:0 frame:0
          TX packets:170 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Pay particular attention to the settings shown for `eth0` and in particular the `Mask`: setting, which should match that which is appropriate for your network. Now check the remaining configuration parameters using `route`:

```
Tycho GPS(root@gsys:~)-> route
```

```
Kernel IP routing table
Destination     Gateway         Genmask           Flags Metric Ref Use Iface
localnet        *               255.255.255.248   U     0      0   0   eth0
loopback        *               255.0.0.0         U     0      0   0   lo
default         192.168.1.241   0.0.0.0           UG    1      0   0   eth0
```

Here you are interested in the default gateway address. It should match the appropriate one for your network. If so, then the ethernet interface of your Tycho has been successfully configured to operate on your network and you are ready to check operation of the Tycho over the network. If not, you should re-check your configuration and/or repeat the **`netconfig`** procedure.

If you have configured a nameserver(s) for your network, you may check that by issuing this shell command:

```
Tycho GPS(root@gsys:~)-> cat /etc/resolv.conf

search your.domain
nameserver 192.168.1.1
nameserver 192.168.1.2
```

Which displays the contents of the */etc/resolv.conf* file containing your domain name and the nameserver IP address(es) to use for that domain.

Check Network Operation

With your Tycho network parameters properly configured, you are ready to test the setup using **`ping`** from a server or workstation that is able to access the network connected to the Tycho. Alternatively, you could **`ping`** one of your servers or workstations from the Tycho shell prompt to test the setup.

Once you have successfully established network communications with the Tycho, you may perform all maintenance and monitoring activities via **`telnet`** and **`ftp`**. The Tycho provides both client and server operation using **`telnet`**. For security reasons as well as to reduce the memory footprint in the Tycho, only client operation is supported using **`ftp`**.

Security conscious users will want to use **`ssh`**, the secure shell replacement for **`telnet`**, as the login means. The companion utility, **`scp`** provides a secure replacement for ftp as a means of transferring files to and from the Tycho. Both of these protocols are supported in the Tycho via the OpenSSH implementations for Linux. Refer to ***Appendix D - Security*** for more information about the secure shell protocol.

Using Telnet

When establishing a **`telnet`** connection with your Tycho, logging in directly as *root* is not permitted. This is a security measure that makes it slightly more difficult to gain access by simply trying passwords, since it is also necessary to know the name of a user. When you initiate a **`telnet`** session with the Tycho, this banner will be displayed:

```
********************************************************************************
*           Welcome to Tycho GPS telnet console on:  gsys.your.domain
********************************************************************************

gsys login:
```

Here you may log in as “gsysuser” with password “Praecis”. When logged in as “gsysuser”, you may check status information and view log files but you will not be able to modify any system settings or view secure files. After correctly entering the password at this prompt,

```
Password:
```

the sign on message is shown. It identifies the host system as Tycho GPS and shows the software part number, version and build date:

```
Tycho GPS 6010-0004-000 v 1.00 Wed May 16 14:17:44 UTC 2002
Tycho GPS(root@gsys:~)->
```

This last line is the standard Tycho shell prompt. The Tycho uses the **bash** shell, which is the Linux standard, full-featured shell. After configuring the unit, you should change the passwords using the **gsyspasswd** command issued from the shell prompt.

To gain *root* access, you must now issue the "super user" command at the shell prompt:

```
Tycho GPS(root@gsys:~)-> su root
```

You will then be prompted for the password, which is "endrun_1", and be granted *root* access to the system. To leave "super user" mode, issue the shell command **exit**. Issuing **exit** again will close the **telnet** session.

Using SSH

When establishing a **ssh** connection with your Tycho, logging in directly as *root* is permitted. When you log in as *root* via a **ssh** session with the Tycho, this banner will be displayed:

```
********************************************************************************
*             Welcome to Tycho GPS SSH console on:  gsys.your.domain
********************************************************************************

root@gsys.your.domain's password:
```

Here you may log in as "root" with password "endrun_1". After correctly entering the password the sign on message is shown. It identifies the host system as Tycho and shows the software part number, version and build date:

```
Tycho GPS 6010-0042-000 v 1.00 Fri Aug 20 14:17:44 UTC 2004
Tycho GPS(root@gsys:~)->
```

This last line is the standard Tycho shell prompt. The Tycho uses the **bash** shell, which is the Linux standard, full-featured shell. After configuring the unit, you should change the passwords using the **gsyspasswd** command issued from the shell prompt.

Issuing **exit** will close the **ssh** session.

Connecting Instruments to the Tycho

Rear-panel mounted BNC jacks provide the means of connecting your equipment to the Tycho. The standard Tycho provides two precision output signals capable of driving properly terminated coaxial cables: 1PPS and AM Code. These two signals are DC-coupled and sourced from Advanced CMOS (ACMOS) drivers which are able to maintain output TTL levels into a 50-ohm load. Care should be taken not to short circuit these outputs or to connect them to other voltage sources.

If your unit is equipped with the optional Alarm Output, it will be available on a rear-panel BNC jack labeled "ALARM". Care should be taken not to directly connect this open-collector output to a voltage source. A series current limiting resistor of at least 1K ohms in value should be used. The pull-up voltage must not exceed 40V.

If your primary application for the Tycho is as a frequency standard and you have not purchased one of the optional higher-stability oscillators, you should consider operating with the console port available to enter the **`gpssta`**t command. (See ***Chapter 3 - Control and Status Commands***.) While using this command you will always know whether the Tycho is currently locked to a GPS signal while you are performing measurements based on its frequency outputs. The holdover frequency accuracy of the standard TCXO will degrade to the $5x10^{-8}$ level fairly quickly following GPS signal loss, depending upon the ambient temperature.

Chapter *Three*

Control and Status Commands

This chapter describes the Tycho control and status commands. The Tycho supports several application-specific commands for performing initialization/setup and for monitoring the performance and status of the unit. You do not need knowledge of Linux commands in order to operate the Tycho. However, the Tycho does support a subset of the standard Linux shell commands and utilities. A wealth of information is available from a variety of sources on Linux. Only the Tycho-specific commands will be described in this chapter. The serial I/O port physical and electrical characteristics are defined as well.

General Linux Shell Operation

You do not need to know Linux in order to operate the Tycho. However, for those interested, the command shell used by the Tycho is the Linux standard: **`bash`**. All commands and file names are case sensitive, which is standard for Unix-like operating systems. If you are unfamiliar with Unix-like operating systems, and you would like to be able to more closely monitor or optimize the performance of your Tycho you should consult either the web

http://www.linuxdoc.org

or good Linux reference books like:

Linux in a Nutshell, Seiver, O'Reilly & Associates, 1999.

Running Linux, Welsh, Dalheimer & Kaufman, O'Reilly & Associates, 1999

to learn the ins and out of the Linux command console.

Available User Commands

COMMAND	FUNCTION
accessconfig	Interactive shell script that guides the user in configuring **`telnet, ssh`** and **`snmpd`** access to the Tycho that is limited to specific hosts. The resulting */etc/hosts.allow* and */etc/hosts.deny* files are saved to the non-volatile FLASH disk. Factory default configuration allows access by all hosts.
antfltmask	Prints the current setting for the Antenna Fault Mask.
cpuopts	Returns the current settings for any installed, user-selectable outputs from the CPU Module. These are 1PPS, AM Code and the optional Programmable TTL Output.
cpuoptsconfig	An interactive script that allows the user to modify the settings for the CPU Module outputs listed above.
cpusertime	Prints the current settings for the optional Serial Time output.
cpusertimeconfig	An interactive script that allows the user to modify the settings for the optional Serial Time output.
eraserootfs_1	Command to erase the UPGRADE root file system FLASH partition. This must be executed prior to loading the new file system image during the Linux upgrade process.
gpscaldelay	Prints the calibration delay to the console.
gpsdynmode	Prints the GPS dynamic mode currently in effect to the console.
gpsrefpos	Prints the GPS reference position to the console.
gpsstat	Prints the GPS subsystem status information to the console.
gpstrkstat	Prints the GPS satellite tracking status to the console.
gpsversion	Prints the GPS firmware and FPGA version information to the console.
gsyshwaddr	Prints the ethernet hardware address, if the ethernet has been configured.
gsysosctype	Prints the installed oscillator type, which is one of: TCXO, MS-OCXO, HS-OCXO or Rubidium.
gsyspasswd	Allows the *root* user to change the password for the two configured users on the Tycho: *gsysuser* and *root*. This script calls the standard Linux **`passwd`** binary and then saves the resulting */etc/shadow* file to the non-volatile FLASH disk.
gsysrootfs	Prints the current root file system image, either TychoGPS_0 (factory default) or TychoGPS_1 (field upgrade) which is running in the Tycho to the console.
gsystimemode	Prints the time mode settings in effect for the timecode or Serial Time output.
gsystimemodeconfig	Interactive shell script that guides the user in configuring the time mode settings for the timecode or Serial Time output. Allows setting to the LOCAL, GPS or UTC timescale and if LOCAL, the setting of the offset to UTC and the Daylight Savings Time (DST) start and stop date/time parameters.

gsysversion	Prints the Tycho application software version information to the console.
help	Prints help for Tycho commands (not Linux).
inetdconfig	Interactive shell script that allows the user to configure the list of protocol servers which are started by the `inetd` server daemon running in the Tycho.
netconfig	Interactive shell script that allows the user to configure the IP network subsystem of the Tycho.
oscctrlstat	Prints the oscillator disciplining parameters.
pluginopts	Returns the board type and settings for any optional buffer boards. (In Tycho products these options are not “plug-in” but must be installed at the factory.)
pluginoptsconfig	An interactive script that allows the user to change the settings on any configurable option board.
setantfltmask	Command to enable or mask the Antenna Fault.
setgpscaldelay	An interactive script that allows the user to change the clock calibration delay.
setgpsdynmode	Allows the user to set the dynamic mode of operation of the GPS subsystem. It may be ON or OFF.
setgpsrefpos	Interactive shell script that prompts the user for an accurate reference position, performs syntax and argument validity checking then passes the position to the GPS subsystem.
setsigfltmask	Command to mask or enable the Signal Loss Fault.
settfomfltlvl	Command to change the TFOM Fault Level.
sigfltmask	Prints the current setting for the Signal Loss Fault mask.
tfomfltlvl	Prints the current setting for the TFOM Fault Level.
updaterootflag	Command to update the flag stored in FLASH that is read by the Linux bootloader at boot time to select operation with either the FACTORY or UPGRADE root file system.
upgradegps	Shell script that facilitates the GPS subsystem firmware upgrade process.
upgradekernel	Shell script that facilitates the Linux kernel firmware upgrade process. Limited applicability. Use with caution.

Detailed Command Descriptions

accessconfig

This command starts an interactive shell script that will allow the root user to configure limitation of `telnet`, `ssh` and `snmp` access to the Tycho. By default, the unit is configured to allow access by all users. If you need to limit `telnet`, `ssh` or `snmp` access, e.g. for security reasons, you must run this script as root from either the RS-232 serial I/O port or from a `telnet` or `ssh` session.

This script modifies these files: */etc/hosts.allow* and */etc/hosts.deny*. These are non-volatilely stored in the FLASH disk */boot/etc* directory. You must re-boot the Tycho after running this script for the changes to take effect.

Set: `accessconfig`
Tycho response: Interactive shell script is started.

antfltmask

This command displays the current setting for the Antenna Fault Mask.

Query: `antfltmask`
Tycho response: `Antenna Fault is ENABLED`

cpuopts

This command displays the current settings for the installed CPU Module outputs.

Query: `cpuopts`
Tycho response:

```
CPU Option 1PPS is installed.
Current setting = 20 microseconds.
CPU Option TIME CODE is installed.
Current Setting = IRIG-B122.
```

cpuoptsconfig

This command starts an interactive shell script that will allow the root user to change the settings of any installed CPU Module output. The user-selectable outputs are 1PPS, AM Code, and the optional Prog TTL and Synthesizer.

Set: `cpuoptsconfig`
Tycho response: Interactive shell script is started.

cpusertime

This command displays the current settings for the optional Serial Time Output.

Query: `cpusertime`
Tycho response:

```
Current Serial Time Output Baud Rate Setting = 9600
Current Serial Time Output Format Setting = Sysplex
Current Serial Time Output Parity Setting = Odd
```

cpusertimeconfig

This command starts an interactive shell script that will allow the root user to change the settings of the optional Serial Time Output. The user-selectable outputs are the format (Sysplex, Truetime, EndRun, EndRunX and NENA), the baud rate (4800, 9600, 19200, 57600) and the parity (ODD, EVEN, or NONE).

Set: `cpusertimeconfig`
Tycho response: Interactive shell script is started.

eraserootfs_1

This command erases the UPGRADE root file system FLASH partition in preparation for performing a Linux subsystem firmware upgrade. See ***Appendix B - Upgrading the Firmwar***e for more information.

Set: `eraserootfs_1`
Tycho response: Erase progress as percent is shown.

gpscaldelay

This command displays the current calibration delay setting. The allowable calibration delay range is +500000 to -500000 nanoseconds.

```
Query:              gpscaldelay
Tycho resposne:     +0 nanoseconds
```

gpsdynmode

This command displays the current GPS subsystem dynamic mode of operation. It has two possible settings: ON or OFF. When it is ON, it is assumed that the Tycho is installed on a moving platform. When it is OFF, it is assumed that the Tycho is installed in a stationary location.

When the dynamic mode is OFF, the Tycho will use its accurate reference position to implement Timing Receiver Autonomous Integrity Monitoring (TRAIM) for the utmost in reliability during any GPS system faults. In addition, single satellite operation is possible once an initial accurate position has been determined.

When the dynamic mode is ON, only a very minimal TRAIM algorithm is in effect because the accurate reference position is not static. In addition, a minimum of four satellites must be visible and only 3-D position fixes are used. When the dynamic mode is ON, the source reported for the accurate reference position by **gpsrefpos** is set to DYN.

```
Query:              gpsdynmode
Tycho response:     OFF
```

gpsrefpos

This command displays the current GPS subsystem reference position. The source of the position, which is one of UNK (unknown), DYN (dynamic), USR (user entered) or AVG (24 hour average of GPS fixes) is displayed first. The WGS-84 latitude and longitude in degrees, minutes, seconds format and the height above the WGS-84 reference ellipsoid in meters follow.

```
Query:              gpsrefpos
Tycho response:
CURRENT REFERENCE POSITION = AVG N38d26m36.11s W122d42m56.50s +00032.5 meters
```

gpsstat

This command allows the user to query the status of the GPS timing subsystem. During normal operation, a Linux daemon (SYSTIMED) polls the GPS timing subsystem every 16 seconds. The results of this poll are used to steer the system clock and are saved to a log file. This command parses and formats the information contained therein and prints this fixed-length string having these fields:

```
LKSTAT TFOM = ? YEAR DOY HH:MM:SS.sssssssss LS LF S N VCDAC SN.R FLTS
```

Where:

LKSTAT   is the tracking status of the engine, either LOCKED or NOTLKD.

TFOM = ? A detailed explanation of TFOM is in ***Appendix A - Time Figure-of-Merit***.

Briefly, TFOM indicates clock accuracy where:

4	time error is < 1 us
5	time error is < 10 us
6	time error is < 100 us
7	time error is < 1 ms
8	time error is < 10 ms
9	time error is > 10 ms, unsynchronized state if never locked to GPS.

YEAR is the year of the UTC timestamp of the most recent information received from the GPS subsystem.

DOY is the day-of-year of the UTC timestamp of the most recent information received from the GPS subsystem..

HH:MM:SS.sssssssss is the hour, minute, second.subsecond UTC timestamp of the most recent information received from the GPS subsystem.

LS is the current number of leap seconds difference between the UTC and GPS timescales (13 at the time of this writing).

LF is the future (at the next UTC midnight) number of leap seconds difference between the UTC and GPS timescales (13 at the time of this writing).

S is the Signal Processor State, one of 0 (Acquiring), 1 (GPS Locking), 2 (GPS Locked).

N is the number of GPS satellites being tracked, 0 to 8.

VCDAC is the 20-bit oscillator Voltage Control DAC word, 0 to 1048575 with larger numbers implying higher oscillator frequency. Typical range is 320000 to 640000.

SN.R is the carrier Signal to Noise Ratio, 0.00 to 99.9, measured in dB in the GPS data rate bandwidth. Typical range is 30 to 45.

FLTS is the fault status, which displays the current summary status of the GPS timing subsystem. The summary status is contained in sixteen bits which are displayed in four hexadecimal characters. Assertion of any of these bits will also be indicated by illumination of the red LED. Each bit of each character indicates the status of a subsystem component:

	Bit 3	Bit 2	Bit 1	Bit 0
Char 0	FLASH Write Fault	FPGA Config Fault	No Signal Time-Out	DAC Control Over-Range
Char 1	Antenna Fault	No Polling Events	Time Input Fault	GPS Comm Fault
Char 2	Not Used	Not Used	Not Used	Not Used
Char 3	Not Used	Not Used	Not Used	Not Used

DAC Control Over-Range: This bit indicates that the electronic frequency control DAC for the oscillator has reached either the high (55000) or low (10000) limit while locked to the GPS signal. Unless the unit is being subjected to out-of-specification environmental conditions, this would indicate that the oscillator frequency has drifted near to the end of life region. This should normally only occur after about ten years of operation. The unit will continue to function until the oscillator frequency finally reaches one of the actual DAC endpoints. The unit should be returned to the factory for oscillator replacement at the customer's convenience. Note: The value referred to here is the upper 16 bits of a 20-bit DAC value.

No Signal Time-Out: This bit indicates that the unit has not been able to acquire a GPS signal for one hour while the Time Figure of Merit has been 9, the unsynchronized condition. This could be due to a variety of reasons. If there are no other faults that could explain the inability to receive a signal, then there could be an or antenna failure or blockage. If the condition persists indefinitely, and a problem with the antenna is not evident, the unit may need to be returned to the factory for repair.

FPGA Config Fault: This bit indicates that the microprocessor was unable to configure the FPGA. This would be a fatal fault and the unit should be returned to the factory for repair .

FLASH Write Fault: This bit indicates that the microprocessor was unable to verify a write to the FLASH non-volatile parameter storage area. This should not ever occur under normal operation. This fault would cause erratic operation at the next power cycling since important parameters could be corrupt. The unit should be returned to the factory for repair.

GPS Comm Fault: This bit indicates that the microprocessor is unable to establish communications with the GPS engine. Please report this fault condition to the factory (1-877-749-3878).

Time Input Fault: This bit indicates that the microprocessor received an erroneous time input from the GPS engine. If the condition persists please report it to the factory (1-877-749-3878).

No Polling Events: This bit indicates that the GPS timing subsystem is not receiving polling request from the Linux subsystem (SYSTIMED daemon). This could be due to a hardware or software failure. If the condition persists after cycling the power to the unit, this is a fatal fault and the unit should be returned to the factory for repair.

Antenna Fault: This bit indicates that the GPS antenna or downlead cable has a fault. It indicates either an over or under current condition. Usually it means that the antenna downlead cable is not plugged into the connector on the rear of the Tycho. If the condition persists after checking the antenna/downlead for obvious faults, this is a fatal fault and the unit should be returned to the factory for repair.

The example response indicates that there has been a period without tracking a GPS signal that exceeded the time-out period, that there was a FLASH Write Fault and that there is an Antenna Fault.

Query: **`gpsstat`**

Tycho response:

```
LOCKED TFOM = 4 2001 092 04:48:56.347916732 13 13 2 7 28605 41.6 008A
```

gpstrkstat

This command displays the current GPS subsystem satellite tracking status. A list of eight satellite numbers is displayed, one for each receiver channel. Satellite number 0 is an invalid number and indicates that no satellite is being tracked on that channel. Valid satellite numbers range from 1 to 32.

Query: `gpstrkstat`
Tycho response: `CURRENT SVs TRKD = 08 11 13 22 31 00 00 00`

gpsversion

This command displays the firmware and hardware versions of the GPS subsystem.

Query: `gpsversion`
Tycho response: `F/W 1.00 FPGA 0202`

gsyshwaddr

This command displays the ethernet hardware address, if the IP network is properly configured. Otherwise it returns nothing.

Query: `gsyshwaddr`
Tycho response: `00:D0:C9:25:78:59`

gsysosctype

This command displays the installed oscillator type. It is one of TCXO, MS-OCXO, HS-OCXO or Rubidium. The standard oscillator is the TCXO.

Query: `gsysosctype`
Tycho response: `Installed Oscillator is TCXO.`

gsyspasswd

This command allows the root user to change the passwords of the two configured users on the system: *root* and *gsysuser*. Arguments passed to **`gsyspasswd`** on the command line are passed verbatim to the real **`passwd`** binary program. When **`passwd`** returns, the resulting modified */etc/shadow* file is copied to the non-volatile */boot/etc* directory.

Query: `gsyspasswd gsysuser`
Tycho response: Passwd interactive utility is started.

gsysrootfs

This command displays the currently booted root file system image. It can be either TychoGPS_0 (factory image) or TychoGPS_1 (field upgrade image). Refer to ***Appendix B - Upgrading the Firmware*** for detailed instructions on performing the upgrade procedure.

Query: `gsysrootfs`
Tycho response: `BOOT_IMAGE=TychoGPS_1`

gsystimemode

This command displays the current time mode settings for any optional timecode or Serial Time output. The time mode setting can be UTC, GPS or Local. The Local Time Offset from UTC and the DST Start/Stop parameters are only valid when the Time Mode is LOCAL. A positive Local Time Offset implies a longitude east of the Greenwich Tycho and that local time is ahead of UTC.

Query: `gsystimemode`
Tycho response:

```
Time Mode = LOCAL
Local Time Offset from UTC = -16 (half hours)
DST Start Month = Apr Sunday = 1st  Hour = 02
DST Stop  Month = Oct Sunday = Last Hour = 02
```

gsystimemodeconfig

This command starts an interactive shell script that will allow the user to configure the time mode of operation of any optional timecode outputs or Serial Time output. Selections are UTC, GPS or Local. *These settings have no effect on the operation of the underlying Linux operating system time. It ALWAYS operates in UTC.*

By default, the unit is configured to operate in LOCAL mode with an offset to UTC of zero and with Daylight Savings Time disabled. If you need to modify this operation, you must run this script as root. Settings made using this command are non-volatile.

Set:	**`gsystimemodeconfig`**
Tycho response:	Interactive shell script is started.

gsysversion

This command displays the firmware version and build date of the Tycho.

Query:	**`gsysversion`**
Tycho response:	

```
Tycho GPS 6010-0042-000 v 2.00 Wed Jan 16 22:38:21 UTC 2004
```

help

This command displays a list of the Tycho commands (not Linux commands). To get help on a particular command you would type **`help`**, followed by the command.

Query:	**`help`**
Tycho response:	Tycho commands are displayed.

Query:	**`help gpsstat`**
Tycho response:	Information specific to the **`gpsstat`** command is displayed.

inetdconfig

This command starts an interactive shell script that will allow the user to configure the list of protocol servers which are started by the **`inetd`** server daemon running in the Tycho. Four protocol servers may be configured: TIME, DAYTIME, and TELNET. By default, the unit is configured to start all of these protocol servers. If you need to disable start-up of some or all of these, e.g. for security reasons, you must run this script as *root* from either the RS-232 serial I/O port or from a **`telnet`** or **`ssh`** session.

This script modifies the */etc/inetd.conf* file, which is non-volatilely stored in the FLASH disk */boot/etc* directory. You must re-boot the Tycho after running this script for the changes to take effect.

Set:	**`inetdconfig`**
Tycho response:	Interactive shell script is started.

netconfig

This command starts an interactive shell script that will allow the user to configure the IP network subsystem of the Tycho. By default, the unit is configured to configure itself using the Dynamic Host Configuration Protocol (DHCP). If you need to set up static IP configuration, you must run this script as *root* from the RS-232 serial I/O port during the installation process. Refer to ***Chapter 2 - Using netconfig to Set Up Your IP*** for details on the use of the command.

This script creates or modifies these files: */etc/HOSTNAME*, */etc/hosts*, */etc/networks, /etc/resolv.conf* and */etc/rc.d/rc.inet1*. All of these are non-volatilely stored in the FLASH disk */boot/etc* directory. You must re-boot the Tycho after running this script for the changes to take effect.

Set: **`netconfig`**
Tycho response: Interactive shell script is started.

oscctrlstat

This command displays the current settings for the oscillator control parameters. These parameters are used to discipline the oscillator. The command formats the data and prints this fixed-length string having these fields:

```
Oscctrlstat = LKSTAT COAST ESTERR MEASERR TIMEDEV AGERATE TAU TEMP
```

Where:

LKSTAT is the GPS subsystem control status, either acquiring, locking or locked.

COAST is the number of seconds in coast mode (unlocked).

ESTERR is the estimated time error when in coast mode in seconds.

MEASERR is the last measured time offset while locked in seconds.

TIMEDEV is the time deviation (TDEV) of measurements in seconds.

AGERATE is the regression computed oscillator ageing rate per day (several hour delay before the first measurements are displayed.).

TAU is the oscillator control loop averaging time constant in seconds.

TEMP is the internal temperature in °C for OCXO and Rubidium oscillators only.

Query: **`oscctrlstat`**
Tycho response:

```
Oscctrlstat = LKD  0 2.72e-09 -2.72e-09 1.23e-09 -0.00e+00   235.2  524332 -999.999
```

pluginopts

This command displays the current settings for any installed option boards.

Query: **`pluginopts`**
Tycho response:

```
Digital Buffer, 2001-0004-012-NC is installed.
Port A Current Setting = 10M PPS.
Port B Current Setting = 5M PPS.
Port C Current Setting = 10 PPS.
Port D Current Setting = 10 PPS.
```

pluginoptsconfig

This command starts an interactive shell script that will allow the root user to change the settings of any installed, user-selectable, option board outputs. The user-selectable outputs are all board-specific. For more information see ***Chapter 4 - Optional Rear-Panel Outputs.***

Set:	**`pluginoptsconfig`**
Tycho response:	Interactive shell script is started.

setantfltmask

This command allows the user to enable or mask the GPS antenna fault. Parameter for this command is either MASKED or ENABLED. Setting this command to MASKED will prevent the antenna fault from creating an alarm condition. Some installations may need to mask this fault due to special antenna situations like splitters or DC blocks that confuse the antenna detection circuit. The factory default setting is ENABLED.

Set:	**`antfltmask MASKED`**
Tycho response:	**`Antenna Fault Mask set to MASKED`**

setgpscaldelay

This command starts an interactive shell script that allows the user to change the clock calibration delay. This setting is used to advance or retard the clock in order to compensate for antenna cable length or other external hardware. Allowable range is +500000 to -500000 nanoseconds.

Set:	**`setgpscaldelay`**
Tycho response:	Interactive shell script is started.

setgpsdynmode

This command accepts a single argument: ON or OFF to allow the user to set the dynamic mode of operation of the GPS subsystem. By default, the unit is configured for static operation, so this setting is OFF. If the Tycho will be mounted on a moving platform, like a ship, then this setting must be changed to ON. The change takes place immediately and is stored non-volatilely.

Set:	**`setgpsdynmode ON`**
Tycho response:	**`GPS Dynamic Mode is ON.`**

setgpsrefpos

This command starts an interactive shell script that will allow the user to set the accurate, reference position of the Tycho. By default, the unit is configured to locate itself using the GPS satellites. In some situations, visibility of the sky is limited and the unit will not be able to determine its position. In this case, the user must determine an accurate WGS-84 position by other means and input it using this command. If you need to set the accurate reference position, you must run this script as root. The changes take place immediately. *If the GPS dynamic mode setting is ON (see* **`gpsdynmode`**/ **`setgpsdynmode`** *commands), then running this script will have no effect.*

In addition to setting a new accurate, reference position, the user can also invalidate an existing one. This will force the Tycho to re-establish a new reference position using the GPS satellite constellation.

Set:	**`setgpsrefpos`**
Tycho response:	Interactive shell script is started.

setsigfltmask

This command allows the user to enable or mask the Signal Loss Fault. Parameter for this command is either MASKED or ENABLED. Setting this command to MASKED will prevent a signal loss fault from creating an alarm condition. Some installations may need to mask this fault when operating the NTP server as a Stratum 2 server. The factory default setting is ENABLED.

Set:	`sigfltmask MASKED`
Tycho response:	`Signal Loss Fault Mask set to MASKED`

settfomfltmask

This command allows the user to change the TFOM Fault Level. This is the threshold at which a signal loss fault will be asserted. See ***Appendix A - Time Figure of Merit*** for more information. By changing the TFOM Fault Level you control the point at which the time error will produce a signal loss fault, which then creates an alarm condition. The factory default setting is 9, which is the maximum TFOM value.

Set:	`settfomfltlvl 6`
Tycho response:	`TFOM Fault Level set to 6`

sigfltmask

This command displays the current setting for the Signal Loss Fault Mask.

Query:	`sigfltmask`
Tycho response:	`Signal Loss Fault is ENABLED`

tfomfltlvl

This command displays the current setting for the TFOM Fault Level.

Query:	`tfomfltlvl`
Tycho response:	`9`

updaterootflag

This command allows the user to update the configuration of the Linux bootloader after a new root file system image has been uploaded to the UPGRADE root file system partition, */dev/rootfs_1* of the Tycho FLASH disk. It may also be used to reset the default back to the FACTORY root file system partition. Refer to ***Appendix B - Upgrading the Firmware*** for detailed instructions for performing the upgrade procedure. One argument is accepted, whose value is either 0 or 1, causing a flag to be set that will indicate to the bootloader which root file system image should be loaded by default. If an argument value of 2 is given, then the currently configured default root file system is shown.

Set:	`updaterootflag 1`
Tycho response:	`UPGRADE is the default root file system.`
Query:	`updaterootflag 2`
Tycho response:	`UPGRADE is the default root file system.`

upgradegps

This script allows the user to upgrade the GPS subsystem firmware. It requires one argument: the path to the binary file to be uploaded to the GPS engine. It issues the commands over the serial port to the GPS subsystem that are needed to start the X-modem file transfer, and then displays the responses from the GPS subsystem to the console. When the X-modem 'C' character appears, indicating that the GPS subsystem is ready to receive the file, you must hit the <ENTER> key, and the transfer will begin. After about one minute, it should complete, at which point you should see the GPS subsystem boot messages appear on the console. From these, you will be able to verify that the firmware was successfully upgraded.

In the example console output below, lines which begin with "---" are generated by the **upgradegps** script. All other lines are from the GPS subsystem, with the exception of the shell message indicating that the process **cat < /dev/arm_user** has been terminated, which is normal. In this example, the 'C' character was received three times before the user hit the <ENTER> key to begin the transfer. The last three lines are the boot messages that are sent by the GPS subsystem as it comes up. The firmware version should match that of the binary file that was uploaded. See ***Performing the GPS Upgrade*** in ***Appendix B - Upgrading the Firmware*** for more information.

Set: **upgradegps /tmp/6010-0020-000.bin**

Tycho response:

```
---When you see the `C` character, hit <enter> to begin the upload.

Waiting for download using XMODEM 128 or XMODEM 1K (both with CRC).
Control X will abort download.
CCC
---Starting file upload, should take about 60 seconds...

/sbin/upgradegps: line 26: 27618 Terminated            cat </dev/arm_user

---You should see the GPS subsystem startup message now.  If not, you
---may need to check your binary file and re-perform the procedure.

Tempus Bootloader 6010-0050-000 v 1.00 - May 28 2004 17:31:05
FW 6010-0020-000 v 1.00 - Aug 18 2004 10:47:41
FPGA 6020-0005-000 v 0202
```

upgradekernel

This script allows the user to change the Linux kernel firmware. It requires one argument: the path to the file to be uploaded to the Tycho. Changing the Linux kernel firmware will enable IPv6 operation and should only be done if you have a requirement for IPv6. See ***Appendix F - IPv6 Information*** and ***Performing the Linux Kernel Upgrade*** in ***Appendix B - Upgrading the Firmware*** for more information.

Set **upgradekernel /tmp/newkernelimage**

Tempus LX response: Interactive shell script is started.

RS-232 Serial I/O Port Signal Definitions

The RS-232 DB9M connector on the rear panel of the Tycho is wired as shown below. In order to connect the Tycho to another computer, a null-modem adapter must be used. The serial cable provided with the shipment is wired as a null-modem adapter and can be used to connect the Tycho to your computer.

Tycho DB9M Pin	Signal Name
1	Not Connected
2	Receive Data (RX)
3	Transmit Data (TX)
4	Data Terminal Ready (DTR)
5	Ground
6	Data Set Ready (DSR)
7	Request To Send (RTS)
8	Clear To Send (RTS)
9	Not Connected

Chapter*Four*

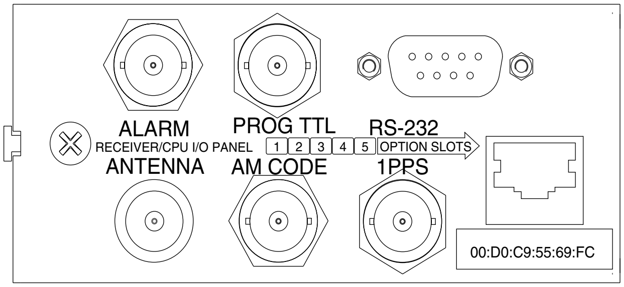
Optional Rear-Panel Outputs

Your Tycho Frequency Reference supports several output options via the CPU module and up to two additional option cards. Status and user settings for the various output signals can be easily viewed and modified via the standard network port or the RS-232 serial port.

CPU Module Options

Standard rear-panel configuration for the CPU module is the Antenna input, the AM Code output, the 1 PPS output, the RS-232 connector and the ethernet connector. Information for the standard outputs, including getting status and control capability via the user interface is described in the main text of this manual. Refer to ***Chapter 3 - Control and Status Commands*** for details on the **`cpuopts`** and **`cpuoptsconfig`** commands.

In addition to the standard connectors, the CPU module can be configured with optional outputs. These optional outputs include a Programmable TTL Output, a Synthesizer Output, an Alarm Output, and a second RS-232 serial port with a Serial Time Output.



Programmable TTL Output

The Programmable TTL Output Option provides user-selectable, on-time pulse rates from 1 PPS to 10 MPPS, or a digital timecode. The output signal can be programmed for any of the various selections via console command **`cpuoptsconfig`**. Refer to Chapter 3 for details on the **`cpuopts`** and **`cpuoptsconfig`** commands. For signal definition see ***Appendix H - Specifications***.

Direct Digital Synthesizer (DDS)

The Programmable TTL Output Option may be upgraded with the addition of the Direct Digital Synthesizer selection. This option provides user-selectable pulse rates from 1 PPS to 10 MPPS, programmable in 1 PPS steps, including 1.544 MPPS or 2.048 MPPS. The selected pulse rate is phase locked to the system oscillator. However, the DDS output is not aligned with system time.

If your Tycho has this option available then it will show up as “SYNTH” in the list of available selections for the Programmable TTL Output. The actual synthesizer frequency is set via console command **`cpuoptsconfig`** (see Chapter 3). For signal definition see ***Appendix H - Specifications***.

Fixed Rate TTL Output (10 MPPS, etc.)

The Fixed Rate Output Option provides an optional customer-specified fixed rate output ranging from 1 PPS to 10 MPPS, or a digital timecode. The rear-panel BNC will be labeled for the appropriate rate such as “10 MPPS” or “100 PPS”, etc. This signal is specified by the customer when the order is placed, preset at the factory, and cannot be changed. There is no user interface status or control. For signal definition see ***Appendix H - Specifications***.

1 PPS (RS-422) Output

This option is provided on a second RS-232 serial port which replaces the two upper BNCs on the CPU Module. The pulse width is normally 1 millisecond wide when shipped from the factory but can be changed via console command **`cpuoptsconfig`**. See signal definition in ***Appendix H - Specifications*** for the 1PPS RS-422 output.

Alarm Output

The Alarm Output Option provides an open-collector output that indicates when the GPS receiver has lost lock, or when serious hardware faults are detected. For a detailed description of the faults see the **`gpsstat`** command in ***Chapter 3 - Control and Status Commands.*** There is no user interface control for this option.

Care should be taken not to directly connect this open-collector output to a voltage source. A series current-limiting resistor of at least 1K ohms in value should be used. The pull-up voltage must not exceed 40V. For signal definition see ***Appendix H - Specifications***.

Serial Time Output

This option is provided on a second RS-232 serial port labeled “Serial Time”. It replaces the two upper BNCs on the CPU Module. It is an output that provides a once-per-second sequence of ASCII characters indicating the current time. The “on-time” character is transmitted during the first millisecond of each second. The output starts automatically on power-up.

To configure this output refer to Chapter 3 for details on the **`cpusertime`** and **`cpusertimeconfig`** commands.

There are several different formats for this string. The format, baud rate and parity can all be changed via console command **`cpusertimeconfig`**. Baud rate selections are 57600, 19200, 9600, and 4800. Parity selections are odd, even, and none. Format selections are Sysplex, Truetime, EndRun, EndRunX and NENA.

Sysplex Format

“Sysplex” means SYStem comPLEX and is a term used to describe computing on clusters of computers. The Sysplex option is designed to provide time synchronization for an IBM Sysplex Timer. It can also be used for precise time synchronization by any computers that do not use NTP and have an available serial port connection. The time contained in this string format is always UTC time. The following string is sent once each second:

<SOH>DDD:HH:MM:SSQ<CR><LF>

<SOH> is the ASCII Start-of-Header character (0x01)

DDD is the day-of-year

: is the colon character (0x3A)

HH is the hour of the day

MM is the minute of the hour

SS is the second of the minute

Q is the time quality indicator and may be either:

<space> ASCII space character (0x20) which indicates locked

? ASCII question mark (0x3F) which indicates the unsynchronized condition

<CR> is the ASCII carriage return character (0x0D) and is the on-time character, transmitted during the first millisecond of each second.

<LF> is the ASCII line feed character (0x0A)

Truetime Format

The format of the Truetime string is identical to the Sysplex format. The only difference between the two is that the Sysplex format always uses UTC time. The time contained in the Truetime format depends on the time mode of the Tycho. (See **`gsystimemodeconfig`** in ***Chapter 4 - Control and Status Commands***.) For example, if you want an output with this string format that uses Local Time, then select the Truetime format.

EndRun Format

The following string is sent once each second:

T YYYY DDD HH:MM:SS zZZ m<CR><LF>

T is the Time Figure of Merit (TFOM) character described in ***Appendix A - TFOM***. This is the on-time character, transmitted during the first millisecond of each second.

YYYY is the year

DDD is the day-of-year

: is the colon character (0x3A)

HH is the hour of the day

MM is the minute of the hour

SS is the second of the minute

z is the sign of the offset to UTC, + implies time is ahead of UTC.

ZZ is the magnitude of the offset to UTC in units of half-hours. Non-zero only when the Timemode is Local (see Chapter 5).

m is the Timemode character and is one of:

G = GPS
L = Local
U = UTC

<CR> is the ASCII carriage return character (0x0D)

<LF> is the ASCII line feed character (0x0A)

EndRunX (Extended) Format

The EndRunX format is identical to the EndRun format with the addition of two fields - the current leap second settings and the future leap second settings. The following string is sent once each second:

```
    T YYYY DDD HH:MM:SS zZZ m<CR><LF>
```

T is the Time Figure of Merit (TFOM) character described in ***Appendix A - TFOM***.
This is the on-time character, transmitted during the first millisecond of each second.

YYYY is the year

DDD is the day-of-year

: is the colon character (0x3A)

HH is the hour of the day

MM is the minute of the hour

SS is the second of the minute

z is the sign of the offset to UTC, + implies time is ahead of UTC.

ZZ is the magnitude of the offset to UTC in units of half-hours.
Non-zero only when the Timemode is Local (see Chapter 5).

m is the Timemode character and is one of:
G = GPS
L = Local
U = UTC

CC is the current leap seconds value.

FF is the future leap seconds value.

<CR> is the ASCII carriage return character (0x0D)

<LF> is the ASCII line feed character (0x0A)

NENA Format

NENA is the National Emergency Number Association. This organization has adopted a format for use in PSAPs (Public Safety Answering Points). This format follows:

```
<CR><LF>Q  DDD HH:MM:SS dTZ=XX<CR><LF>
```

Q is the time quality indicator and may be either:
<space> ASCII space character (0x20) which indicates locked.
? ASCII question mark (0x3F) which indicates the unsynchronized condition.

DDD is the day-of-year

: is the colon character (0x3A)

HH is the hour of the day

MM is the minute of the hour

SS is the second of the minute

d is the DST indicator (S,I,D,O).

TZ=XX is the time zone where XX is 00 through 23

<CR> is the ASCII carriage return character (0x0D).
The first <CR> is the on-time character.

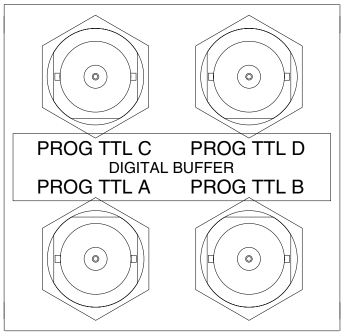
<LF> is the ASCII line feed character (0x0A)

Module Options

Your Tycho Frequency Reference has two available option slots that can be configured with a variety of modules. (In the Tycho these modules are not field-installable - they must be installed at the factory.) Refer to `pluginopts` and `pluginoptsconfig` in ***Chapter 3 - Control and Status Commands***. Option modules are described below:

Programmable Digital Buffer Module

The Programmable Digital Buffer Module adds four independently programmable TTL outputs to your Tycho. These buffered outputs can provide on-time pulse rates from 1 PPS to 10 MPPS, or a digital timecode output. Each output can be individually programmed for any of the various selections via console command `pluginoptsconfig` (see Chapter 3). For signal definition see ***Appendix H - Specifications***. The rear panel outputs are identified as PROG TTL A through D as shown below:



Digital Timecode Formats

If you have selected timecode as the output on the Digital Buffer Module then its format will be the same format as that on the standard CPU Module AM Code connector. This format can be changed via console command `cpuoptsconfig` (see Chapter 3).

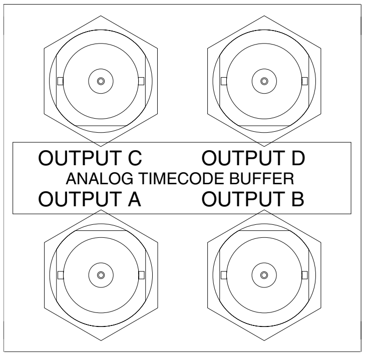
Direct Digital Synthesizer (DDS)

An additional upgrade to the Programmable Digital Buffer Module provides access to the DDS that is resident on the Tycho CPU module. Programmable synthesized pulse rates from 1 PPS to 10 MPPS in 1 PPS steps are available, including 1.544 MPPS or 2.048 MPPS. The pulse rate is phase locked to the system oscillator. However, the DDS output is not aligned with system time.

If your Tycho has this option available then it will show up as “SYNTH” in the list of available selections for the Programmable Digital Buffer outputs. The actual synthesizer frequency that is resident on the Tycho CPU module is set via console command `cpuoptsconfig` (see Chapter 3).

Analog Timecode Buffer Module

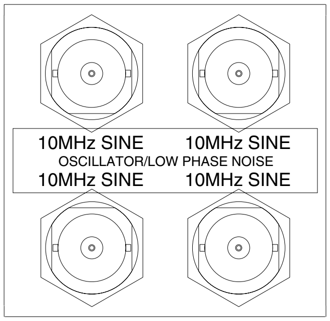
The Analog Timecode Buffer Module can provide synchronization of equipment such as synchronized generators, digital fault recorders, SCADA systems, and time displays, and are suitable for recording onto magnetic tape or for transmission over another medium such as coaxial cable. Available timecode formats are: IRIG-B120 (IEEE-1344), IRIG-B122, IRIG-B123, NASA-36, or 2137. All four output connectors use the same code format. The format is identical to that of the CPU Module AM Code output which is set via console command `cpuoptsconfig` (see Chapter 3). For signal definition see ***Appendix H - Specifications***.



Oscillator/Low Phase Noise Module

The Tycho can be configured with several high-performance, disciplined, 10-MHz oscillators. The Low Phase Noise Output Option works in conjunction with these oscillators to provide up to eight individually buffered, spectrally pure, sinewave outputs. The levels of the contributors to spectral impurity have been carefully controlled by the selection of the optional oscillators that are offered, and by the design of the option module and its integration into the Tycho chassis. In addition, very good channel-to-channel isolation has been achieved.

This module always resides in Slot 1 (next to the CPU module). The rear-panel outputs are labeled with their configured frequency. These frequencies are not changeable but can be viewed via console command `pluginopts` (see Chapter 3). For signal definition see ***Appendix H - Specifications***.



Appendix A

Time Figure-of-Merit (TFOM)

This appendix describes the Time Figure of Merit (TFOM) number. The Tycho displays this number in the time-of-day fields printed by the Tycho **`gpsstat`** *command (see Chapter 3). The TFOM number indicates the level of accuracy that should be included in the interpretation of the time-of-day and ranges from 4 to 9:*

4	time error is < 1 us
5	time error is < 10 us
6	time error is < 100 us
7	time error is < 1 ms
8	time error is < 10 ms
9	time error is > 10 ms, unsynchronized state if never locked to GPS

In all cases, the Tycho reports this value as accurately as possible, even during periods of GPS signal outage where the Tycho is unable to directly measure the relationship of its timing outputs to UTC. During these GPS outage periods, assuming that the Tycho had been synchronized prior to the outage, the Tycho extrapolates the expected drift of the Tycho timing signals based on its knowledge of the characteristics of the internal Temperature Compensated Crystal Oscillator (TCXO), Oven Controlled Crystal Oscillator (OCXO) or Rubidium oscillator. The extrapolated TFOM is based on a conservative estimate of the performance of the oscillator and should be considered ‘worst case’ for a typical benign ambient temperature environment.

Due to this extrapolation behavior, after initial synchronization, brief periods without GPS satellite visibility will not induce an immediate alarm condition. (Removal of the antenna to simulate this will induce an immediate alarm, however.) If the condition persists for long enough periods, you should see the TFOM character change to indicate a gradually deteriorating accuracy of the timing outputs. If the signal loss condition persists longer, then the final, unsynchronized state will eventually be reached. If the Tycho is unable to achieve re-synchronization within one hour after reaching this state, the red LED will illuminate. The fault status field returned in the **`gpsstat`** command will have the appropriate bit set to indicate a loss-of-signal time-out condition.

Appendix *B*

Upgrading the Firmware

Periodically, EndRun Technologies will make bug fixes and enhancements to our products available for download from our website. All such downloads are freely available to our customers, without charge. After you have downloaded the appropriate FLASH binary image file from the EndRun Technologies website, you are ready to perform the upgrade to your Tycho.

The firmware consists of two FLASH binary image files. One of these is the firmware for the Tycho itself. This firmware executes on the IBM-compatible CPU and contains the embedded Linux operating system. The other file is the firmware for the GPS time and frequency subsystem. Each of these files may be upgraded independently, although some upgrades require both images to be modified together.

What You Need To Perform the Upgrade

You will need to use **`ftp`** or **`scp`** to transfer the binary image file(s) to the Tycho. This means that you must place the previously downloaded file(s) in a place on your network which is accessible to the Tycho.

Performing the Linux Upgrade

There are two FLASH disk partitions which hold the compressed Linux root file system images. These partitions are raw FLASH blocks, have no file system and may not be mounted. They are accessed through low-level devices. To protect the factory root file system from accidental erasure or over-writing, the device node has been deleted. The upgrade FLASH disk partition is accessed via */dev/rootfs_1*. When performing an upgrade, you will be copying the new image to this device.

CAUTION

Some browsers will automatically unzip the gzip file when downloading from the website. Please make sure that the gzip file is less than 6M in size before proceeding. Upgrading the partition with a too-large file size can cause serious problems and the unit may have to be returned to the factory for re-programming.

To perform the upgrade, log in as the *root* user to the Tycho using the local console serial I/O port, **`telnet`** or **`ssh`** and perform these operations:

First erase the upgrade partition by issuing this command at the shell prompt:

```
eraserootfs_1
```

If you are using **ftp** to perform the upgrade, transfer the previously downloaded file using *binary* transfer mode from the remote host to */dev/rootfs_1* on your Tycho using FTP (substitute the name of the root file system image that you are installing for *rootfsupgrade.gz*): Issue these commands from the console of your Tycho:

```
ftp remote_host                         {perform ftp login on remote host}
bin                                     {set transfer mode to binary}
get rootfsupgrade.gz /dev/rootfs_1      {transfer the file}
quit                                    {close the ftp session after transfer }
```

If you are using **ssh**, you may open a command window on the remote computer and securely transfer the root file system image using **scp** from the remote computer to your Tycho. A command like this should be used:

```
scp -p rootfsupgrade.gz root@gsys.your.domain:/dev/rootfs_1
```

Update the default file system partition by issuing this command on your Tycho.

```
updaterootflag 1
```

You should see this line displayed:

```
UPGRADE is the default root file system.
```

Now reboot the system by issuing this command at the shell prompt:

```
shutdown -r now
```

Wait about 30 seconds for the system to shutdown and re-boot. Then log in to the Tycho using **telnet** or **ssh**. If all has gone well, you should be able to log in the usual way. After you have entered your password, the system message will be displayed. You should notice that it now indicates the software version and date of the upgrade that you previously downloaded. You can also check this at any time by issuing

```
gsysversion
```

which will cause the system message to be re-displayed.

You can also check to see which root file system image the system is currently booted under by issuing this command at the shell prompt:

```
gsysrootfs
```

Which should cause this to be printed to the console:

```
BOOT_IMAGE=TychoGPS_1
```

If so, and your unit seems to be operating normally, you have successfully completed the upgrade. If your unit does not boot up successfully, and you are not able to **telnet** or **ssh** into the system after 30 seconds, then there has been some kind of problem with the upgrade. It is possible that the file downloaded was corrupt or that you forgot to set your FTP download file mode to binary when downloading the file--either from the EndRun Technologies website or when transferring it to the Tycho.

Recovering from a Failed Upgrade

To restore your Tycho to a bootable state using the factory root file system, you must use the serial I/O port and re-boot the Tycho by cycling the power. For setup details refer to sections ***Connect the Serial I/O Port*** and ***Test the Serial I/O Port*** in ***Chapter 2 - Basic Installation***. When you have connected your terminal to the serial I/O port, apply power to the Tycho.

Pay close attention to the terminal window while the unit is re-booting. After the Linux bootloader displays the message

```
To override and boot the FACTORY partition type 'FACTORY' within 5 seconds...
```

you must begin typing "factory" within five seconds to let the bootloader know that you are going to override the default root file system. After you hit <enter> the bootloader will boot the factory root file system. Watch the rest of the boot process to make sure that you have successfully recovered. If the system boots normally, then you should resolve the problems with the previous upgrade and re-perform it.

Performing the Linux Kernel Upgrade

If you want to upgrade your kernel to the IPv6-capable one then you must first be sure that your root file system is version 2.60 or later.

To upgrade your kernel, log in as the *root* user to the Tycho using the local console serial I/O port, **telnet** or **ssh** and perform these operations:

If you are using **ftp** to perform the upgrade, transfer the previously downloaded file using *binary* transfer mode from the remote host to a temporary location on your Tycho using FTP (substitute the name of the kernel image that you are installing for *newkernelimage*): Issue these commands from the console of your Tycho:

```
ftp remote_host                        {perform ftp login on remote host}
bin                                    {set transfer mode to binary}
get newkernelimage /tmp                {transfer the file}
quit                                   {close the ftp session after transfer }
```

If you are using **ssh**, you may open a command window on the remote computer and securely transfer the root file system image using **scp** from the remote computer to your Tycho. A command like this should be used:

```
scp -p newkernelimage root@cntp.your.domain:/tmp
```

The kernel upgrade utility is executed with a single argument passed on the command line: the path to the previously uploaded kernel image file. For example:

```
upgradekernel /tmp/newkernelimage
```

The kernel upgrade utility verifies the integrity of the file, reads the kernel version information, presents it to you and asks you to verify before replacing the old kernel image. If you verify, it will then erase the old image and write the new one in its place. The erase and write operation takes about 10 seconds.

CAUTION

A power failure during the kernel erase and write operation would render your unit unbootable. It is highly advisable to plug your unit into a UPS while performing the kernel upgrade.

Performing the GPS Upgrade

To perform this upgrade, log in as the *root* user to the Tycho using either the local console serial I/O port, **telnet** or **ssh** and perform these operations:

Change the working directory to the */tmp* directory:

```
cd /tmp
```

If you are using **ftp** to perform the upgrade, transfer the previously downloaded file using *binary* transfer mode from the remote host to the working directory, */tmp* (substitute the name of the GPS subsystem image that your are installing for *gpsupgrade.bin*):

```
ftp remote_host         {perform ftp login on remote host}
bin                     {set transfer mode to binary}
get gpsupgrade.bin      {transfer the file}
quit                    {close the ftp session after the transfer }
```

If you are using **ssh**, you may open another command window on the remote computer and securely transfer the GPS subsystem image to the */tmp* directory using **scp** from the remote computer. A command like this could be used:

```
scp -p gpsupgrade.bin root@gsys.your.domain:/tmp
```

Now issue the following command to the Tycho console to initiate the upload:

```
upgradegps /tmp/gpsupgrade.bin
```

This command is a script that performs the file transfer to the GPS engine. It first tells the GPS engine to enter the ‘waiting for download’ mode, and then prompts you with this line

```
---When you see the `C` character, hit <enter> to begin the upload.
```

Then it echos the serial port characters sent by the GPS engine to the console. You should next see this message from the GPS engine:

```
Waiting for download using XMODEM 128 or XMODEM 1K (both with CRC).
Control X will abort download.
```

After about 3 seconds, you should see a capital ‘C’ character appear. When you do, hit the <enter> key. Now the script will initiate the XMODEM file transfer and display this message to the console:

```
---Starting file upload, should take about 60 seconds...
```

After about one minute you should see this message from the script:

```
/sbin/upgradegps: line 26: 27618 Terminated       cat </dev/arm_user

---You should see the GPS sub-system startup message now.  If not, you
---may need to check your binary file and re-perform the procedure.
```

The first message should be ignored. It is only reporting that one of the intermediate processes of the script execution has been terminated. The next message informs you that the GPS engine file transfer has completed, and that its start-up messages should appear. First the bootloader message will appear:

```
Tempus Bootloader 6010-0050-000 v 1.00 - May 28 2004 17:31:05
```

In about ten seconds, the GPS engine application start-up messages should appear:

```
FW 6010-0020-000 v 1.00 - Aug 18 2004 10:47:41
FPGA 6020-0005-000 v 0202
```

The firmware version should match that of the binary file that you uploaded. At this point, the **upgradegps** script terminates its execution, and you will again have the standard Tycho console prompt.

After about one minute, you should query the GPS firmware version using the command:

```
gpsversion
```

The upgraded version information should be displayed.

Problems with the GPS Upgrade

Should you have difficulties with the upgrade due to a corrupt file, power failure during upload, or other accident, do not be alarmed. Even though you may have lost the existing application program, the GPS engine bootloader program will remain intact. On boot up, it will check to see if a valid application program is in the FLASH memory. If there is not, it will immediately go into the ‘waiting for download’ mode. You may verify this by issuing this command:

```
cat < /dev/arm_user
```

You should now see the ‘C’ character being received every three seconds. This is the character that the GPS engine bootloader sends to indicate to the XMODEM utility that it is wating for a download. You may now re-try the upload procedure, assuming that you have corrected any original problem with the binary file. First kill the **cat** command by typing CTRL-C. You should see a command prompt. Now issue this command to re-transfer the binary file:

```
upgradegps /tmp/gpsupgrade.bin
```

Appendix C

Simple Network Management Protocol (SNMP)

Your Tycho includes the (NET)-SNMP version 5.3.1 implementation of an SNMP agent, **`snmpd`***, and a SNMP notification/trap generation utility,* **`snmptrap`***. It supports all versions of the protocol in use today: SNMPv1 (the original Internet standard), SNMPv2c (never reached standard status, often called "community SNMP") and SNMPv3 (the latest Internet standard).*

The NET-SNMP project has its roots in the Carnegie-Mellon University SNMP implementation. For more detailed information about the NET-SNMP project and to obtain management software and detailed configuration information, you can visit this website: http://www.net-snmp.org .

An excellent book which describes operation and configuration of various SNMP managers and agents, including the NET-SNMP implementations, is available from O'Reilley & Associates:

Essential SNMP, Mauro & Schmidt, O'Reilley & Associates, 2001

If you are planning to operate with SNMPv3, it is highly recommended that you make use of both of these resources to familiarize yourself with the agent configuration concepts.

SNMPv3 Security

Prior to SNMPv3, SNMP had definite security inadequacies due to using two community names in a manner analogous to passwords that were transmitted over the network as clear text. In addition, since no mechanism existed for authenticating or encrypting session data, any number of man-in-the-middle data corruption/replacement exploits were possible in addition to plain old snooping to learn the community names. SNMPv3 implements the User-based Security Model (USM) defined in RFC-2274 which employs modern cryptographic technologies to both authenticate multiple users and to encrypt their session data for privacy, much in the same way that SSH does for remote login shell users.

In addition, it implements the View-based Access Control Model (VACM) defined in RFC-2275. This RFC defines mechanisms for limiting the access of multiple users having various security levels (no authentication, authentication or authentication plus privacy) to specific "views" of the Structure of Management Information (SMI) object tree.

Enterprise Management Information Base (MIB)

In addition to providing the SNMP variables contained in MIB-II as described in RFC-1213, EndRun Technologies has implemented an enterprise MIB using the syntax of the SMI version 2 (SMIv2) as described in RFC-2578:

TYCHO-MIB

Which is located on your Tycho in this ASCII file:

/usr/local/share/snmp/mibs/TYCHO-MIB.txt

In addition to a complete set of GPS status objects, the MIB defines two SMIv2 notification objects:

- GPS Fault Status change
- GPS Time Figure of Merit change

Invocation of the SNMP daemon

The SNMP daemon, **snmpd** is started from the */etc/rc.d/rc.local* system start-up script with this line:

```
snmpd -m “MIBNAME” -Ls -d -c /etc/snmpd.conf
```

By default, it will listen on port 161 for SNMP queries from the network management system. If you would like to have it listen on another port, you could edit the file by adding **-p port** to the end of this line, where **port** is the number of the port you would like for the agent to listen on. If you would like to disable starting of the **snmpd** daemon altogether, you can either remove this line or place a **#** character at the beginning of the line so that it will not be executed. (A very compact editor with WordStar command keystrokes is available on the system for this purpose: **edit**. If you start **edit** without giving it a file name to open, it will display its help screen, showing the supported keystrokes.)

IMPORTANT

After editing ***/etc/rc.d/rc.local***, you must copy it to the ***/boot/etc/rc.d*** directory and re-boot the system. It is very important to retain the access mode for the file, so be sure to use **cp -p** when performing the copy. During the boot process, the files contained in the ***/boot/etc/rc.d*** directory are copied to the working ***/etc/rc.d*** directory on the system RAM disk. In this way the factory defaults are overwritten.

Quick Start Configuration -- SNMPv1/v2c

You should be able to compile the TYCHO-MIB file on your SNMP management system and access the variables defined therein. The factory default community names are “TychoGPS” for the read-only community and “endrun_1” for the read-write community. This is all that is required for operation under v1 and v2c of SNMP. You can, and should, change the default community names by editing */etc/snmpd.conf* and modifying these two lines:

```
rwcommunity   endrun_1
rocommunity  TychoGPS
```

Configuring SNMPv1 Trap Generation

To have your Tycho send SNMPv1 traps (RFC-1215) you must configure the community and destination for SNMPv1 traps by uncommenting and editing this line in */etc/snmpd.conf*:

```
trapsink    xxx.xxx.xxx.xxx trapcommunity trapport
```

where **`trapcommunity`** should be replaced by your community, and **`xxx.xxx.xxx.xxx`** is the IP address or hostname of the destination host for receiving the traps generated by the Tycho. By default, the trap will be sent to port 162. You may optionally add another parameter, **`trapport`** to the end of the above line to override the default port setting. Otherwise leave it blank.

Note: Though the agent will recognize multiple **`trapsink`** lines within */etc/snmpd.conf* and send the generic SNMP coldStart or authenticationFailure traps to each destination, the enterprise trap generation mechanism of the Tycho will only send a trap to the last declared **`trapsink`** in the file.

Configuring SNMPv2c Notifications and Informs

To have your Tycho send SNMPv2c notifications (SMIv2, RFC-2578) or informs, you must configure the communities and destinations by uncommenting and editing one or both of these lines in */etc/snmpd.conf*:

```
trap2sink    xxx.xxx.xxx.xxx trap2community trap2port
informsink   xxx.xxx.xxx.xxx informcommunity informport
```

where **`trap2community`** and **`informcommunity`** should be replaced by your communities, and **`xxx.xxx.xxx.xxx`** is the IP address or hostname of the destination host for receiving the notifications or informs generated by the Tycho. By default, the v2c trap or inform will be sent to port 162. You may optionally add another parameter, **`trap2port`** or **`informport`** to the ends of the above lines to override the default port setting. Otherwise leave it blank.

Note: Though the agent will recognize multiple **`trap2sink`** or **`informsink`** lines within */etc/snmpd.conf* and send the generic SNMP coldStart or authenticationFailure notifications and informs to each destination, the enterprise notification/inform generation mechanism of the Tycho will only send a notification to the last declared **`trap2sink`** and an inform to the last declared **`informsink`** in the file.

IMPORTANT

After editing ***/etc/snmpd.conf***, you must copy it to the ***/boot/etc*** directory and re-boot the system. It is very important to retain the access mode for the file (readable only by ***root***), so be sure to use **`cp -p`** when performing the copy. During the boot process, the files contained in the ***/boot/etc*** directory are copied to the working ***/etc*** directory on the system RAM disk. In this way the factory defaults are overwritten.

Configuration of SNMPv3

If you are planning to use SNMPv3, you should definitely make use of the two resources mentioned previously (NET-SNMP website and *Essential SNMP*) and study them carefully. There are rather elaborate configuration options available when you are using v3. The instruction presented here will give you the flavor of the configuration but definitely not the full scope of possibilities. To access your Tycho via v3 of SNMP, you will have to configure two files:

/etc/snmpd.conf
/boot/net-snmp/snmpd.conf

The first file contains static configuration parameters that the agent uses to control access and to determine where to send notifications/traps. Other aspects of the agent's operation are also configurable in this file, but you should not need to modify those. To use the SNMPv3 capabilities of the Tycho, you must first set up user information and access limits for those users in */etc/snmpd.conf.* Uncomment and edit these two lines to define your v3 users and their access parameters:

```
rwuser root    priv .1
rouser tychouser auth .1.3.6.1.4.1.13827
```

The first line defines a SNMPv3 read-write user *root* whose minimum security level will be authenticated and encrypted for privacy (choices are noauth, auth and priv), and who will have read-write access to the entire *iso(1)* branch of the SMI object tree. The second line defines a SNMPv3 read-only user *tychouser* whose minimum security level will be authenticated but not encrypted, and who will have read-only access to the entire *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).endRunTechnologiesMIB(13827)* branch of the SMI object tree. After adding the user lines to */etc/snmpd.conf*, copy it to the */boot/etc* directory using **`cp -p`**.

The second file is located on the non-volatile FLASH disk and is used by the SNMP agent to store "persistent data" that may be dynamic in nature. This may include the values of the MIB-II variables sysLocation, sysContact and sysName as well as any configured SNMPv3 user crypto keys. In order to use SNMPv3, you must configure user keys in this file for each SNMPv3 user that you have set up in */etc/snmpd.conf*. To do this, you must add lines to */boot/net-snmp/snmpd.conf* like these for each user:

```
createUser root    MD5 endrun_1 DES endrun_1
createUser tychouser SHA TychoGPS
```

The first line will cause the agent, **`snmpd`** to create a user *root* who may be authenticated via Message Digest Algorithm 5 (MD5) with password *endrun_1* and may use the Data Encryption Standard (DES) to encrypt the session data with passphrase *endrun_1*. The second line will cause a user *tychouser* to be created who may be authenticated using the Secure Hash Algorithm (SHA) with password *TychoGPS_0*. Passwords and passphrases must have a *minimum* of 8 characters, or you will not be able to be authenticated.

IMPORTANT

You must kill the **`snmpd`** process prior to editing, ***/boot/net-snmp/snmpd.conf***. Otherwise, the secret key creation may not complete properly. Issue the command **`ps -e`** to have the operating system display the list of running processes. Look for the PID of the **`snmpd`** process and issue the kill command to stop it. For example, if the PID listed for the **`snmpd`** process is 53, then you would issue this command: **`kill 53`**. You can verify that the process was terminated by re-issuing the **`ps -e`** command.

After re-booting, the agent will read the */boot/net-snmp/snmpd.conf* configuration file and compute secret key(s) for each of the users and delete the **`createUser`** lines from the file. It will then write the secret key(s) to the file. These lines begin with the string, **`usmUser`**. In this way, un-encrypted passwords are not stored on the system.

IMPORTANT

To generate new keys, stop the **`snmpd`** process, delete the existing **`usmUser`** key lines from the file ***/boot/net-snmp/snmpd.conf*** and then add new **`createUser`** lines. Then re-boot the system.

This example gives the simplest configuration to begin using SNMPv3 but doesn't make use of the full capabilities of the VACM in defining groups and views for fine-grained access control. The factory default */etc/snmpd.conf* file contains commented blocks of lines that can be uncommented to give you a basic configuration that uses the User-based Security Model (USM) described in RFC-2274 and the View-based Access Control Model (VACM) described in RFC-2275. The comments included in the file should help you in modifying it for your specific requirements.

Appendix *D*

Security

Your Tycho Frequency Reference incorporates several important security features to prevent unauthorized tampering with its operation. Many of these are standard multiple-user access control features of the underlying Linux operating system which controls the Tycho. Others are provided by the additional protocol servers selected for inclusion in your Tycho, and the way that they are configured.

Secure user authentication and session privacy while performing routine monitoring and maintenance tasks are provided by the OpenSSH implementations of the "secure shell" daemon, **`sshd`** *and its companion "secure copy" utility,* **`scp`**. *The NET-SNMP implementation of the Simple Network Management Protocol (SNMP) daemon,* **`snmpd`**. *conforms to the latest Internet standard, known as SNMPv3, which also supports secure user authentication and session privacy. This appendix describes these security measures and gives the advanced network administrator information that will allow custom configuration to fit specific security needs.*

Linux Operating System

The embedded Linux operating system running in the Tycho is based on kernel version 2.4.26 and version 10 of the Slackware Linux distribution. As such it supports a complete set of security provisions:

- System passwords are kept in an encrypted file, */etc/shadow* which is not accessible by users other than *root*.

- Direct *root* logins are only permitted on the local RS-232 console or via SSH.

- The secure copy utility, **`scp`**, eliminates the need to use the insecure **`ftp`** protocol for transferring program updates to the Tycho.

- Access via SNMP is configurable to provide the security of the latest version 3 Internet standard which supports both view-based access control and user-based security using modern encryption techniques. Previous versions v1 and v2c supported access control essentially via passwords transmitted over the network in plain text. Refer to ***Appendix C – Simple Network Management Protocol*** which is dedicated to configuration of SNMP for details.

- Individual host access to protocol server daemons such as **`in.telnetd, snmpd`** or **`sshd`** may be controlled by the **`tcpd`** daemon and */etc/hosts.allow* and */etc/hosts.deny.*

- Risky protocols like TIME, DAYTIME and TELNET may be completely disabled by configuration of the **`inetd`** super-server daemon.

The last two topics are supported on the Tycho by a pair of shell scripts which ease configuration for the inexperienced user of Unix-like operating systems. These are **`accessconfig`** and **`inetdconfig`**.

accessconfig modifies two files which are used by **tcpd** and the standalone daemons, **snmpd** and **sshd** to determine whether or not to grant access to a requesting host: */etc/hosts.allow* and */etc/hosts/deny*. These two files may contain configuration information for a number of protocol servers, but in the Tycho only access control to the protocol server daemons **in.telnetd, sshd** and **snmpd** is configured.

As shipped from the factory, these two files are empty. When the user runs **accessconfig**, these lines are added to the */etc/hosts.deny* file:

```
in.telnetd:  ALL
sshd:  ALL
snmpd:  ALL
```

This tells **tcpd** to deny access to **in.telnetd** and **sshd** to all hosts not listed in the */etc/hosts.allow* file. The **snmpd** and **sshd** daemons also parse this file prior to granting access to a requesting host. Then the user is prompted to enter a list of hosts that will be granted access to **in.telnetd, sshd** and **snmpd**. These appear in the */etc/hosts.allow* as lines like this:

```
in.telnetd:  192.168.1.2, 192.168.1.3
sshd:  192.168.1.2, 192.168.1.3
snmpd:  192.168.1.2, 192.l68.1.3
```

This simple shell script handles the needs of most users, however the syntax of these two files supports elaborate configuration possibilities which are beyond the capabilites of this simple shell script. Advanced users who need these capabilities will need to edit these two files directly and then copy them to the */boot/etc* directory. (A very compact editor with WordStar command keystrokes is available on the system for this purpose: **edit**. If you start **edit** without giving it a file name to open, it will display its help screen, showing the supported keystrokes.) Be careful to maintain the proper ownership and access permissions by using **cp -p** when copying the files.

inetdconfig modifies the */etc/inetd.conf* file which is read by **inetd** to start-up various protocol server daemons when requests from remote hosts are received. Currently, three servers are configurable via **inetdconfig**: TIME and DAYTIME, whose daemons are contained within the **inetd** daemon itself, and **in.telnetd**. Any one or all of these may be enabled or disabled for start-up.

OpenSSH

The secure shell protocol server running in the Tycho is based on the portable OpenSSH for Linux. As such it supports both SSH1 and SSH2 protocol versions. For more information about this protocol and to obtain client software, refer to the OpenSSH website: http://www.openssh.com.

An excellent book which describes operation and configuration of the various SSH implementations, including OpenSSH is available from O'Reilley & Associates:

SSH, The Secure Shell, Barrett & Silverman, O'Reilley & Associates, 2001

In the interest of conserving scarce system memory resources, only the secure shell server daemon, **sshd** and the secure copy utility, **scp**, are implemented in the Tycho. This means that users on remote hosts may log in to the Tycho via an **ssh** client, but users logged in on the Tycho are unable to log in to a remote host via **ssh**. Since **scp** runs in concert with an **ssh** client, the same limitations exist for its use, i.e. users on remote hosts may transfer files to and from the Tycho via **`scp`** over **`ssh`** but users logged in on the Tycho are unable to transfer files to and from a remote host via **`scp`** over **`ssh`**.

The factory configuration contains a complete set of security keys for both SSH1 and SSH2 versions of the protocol. RSA keys are supported by both versions, and DSA keys are supported when using the SSH2 version.

In addition, the Tycho is factory configured with a set of public keys for passwordless, public key authentication of the root user. To use this capability, the corresponding set of private keys for each of the two SSH versions are provided in the */boot/root* directory of the Tycho. Three files contain these keys: *identity* (SSH1), *id_rsa* (SSH2) and *id_dsa* (SSH2). These must be copied to the user's *~/.ssh* directory on their remote computer. (Be careful to maintain the proper ownership and access permissions by using **`cp -p`** when copying the files. They MUST be readable only by *root*.) The corresponding public keys are by factory default resident in the */root/.ssh* directory of the Tycho. Two files contain these keys: *authorized_keys* (SSH1) and *authorized_keys2* (SSH2).

Since the provided private keys are not passphrase protected, the user should create a new set of keys after verifying operation with the factory default key sets. After creating the new keys, the public keys should be copied to the */boot/root/.ssh* directory of the Tycho. At boot time, the Tycho will copy these to the actual */root/.ssh* directory of the system ramdisk, thereby replacing the factory default set of public keys.

Advanced users wishing to modify the configuration of the **`sshd`** daemon should edit the */etc/sshd_config* file and then copy it to the */boot/etc* directory of the Tycho. Be careful to maintain the proper ownership and access permissions by using **`cp -p`** when copying the file. At boot time, it will be copied to the */etc* directory of the system ramdisk, thereby replacing the factory default configuration file.

Appendix *E*

Timecode Formats

A standard feature of your Tycho Frequency Reference is a single timecode output available at the rear panel BNC connector identified as AM CODE. A DC-shift timecode output is available via the optional Programmable TTL Output on the CPU Module.

The output code format is selectable via a console command. See **`CpuOpts`** *in* ***Chapter 3 - Control and Status Commands****. Each format is described below. Time codes are commonly used to provide time information to external devices such as displays, magnetic tape devices, strip chart recorders and several types of embedded computer peripheral cards.*

IRIG-B122/002

This is the most widely used format and is normally the factory default for the AM Code output. The IRIG-B122 format is a 100 pps code and is used to amplitude modulate a 1000 kHz sine wave carrier. The information contained in the timecode is seconds through day-of-year coded in Binary Coded Decimal (BCD). Reference IRIG Document 104-60.

IRIG-B123/003

In addition to the time information identified in B122 above, this format also contains Straight Binary Seconds (SBS) of day. SBS is provided at the end of the frame, in the 17 bits starting in position 80.

IRIG-B120/000 (IEEE-1344-1995)

In addition to the time data and the Straight Binary Seconds data this format provides for time/status data in the control bit positions of IRIG-B. The information provided there is defined by IEEE standard 1344-1995: Unit and Tens of Years, Leap Second, Daylight Savings, Local Time Offset, Time Quality and Parity. The IEEE-1344 table provided below shows each bit position with detailed information.

NASA-36

NASA-36 bit time code is a 100-bit, pulse width modulated format used to amplitude modulate a 1000 kHz sine wave carrier. The information contained in the timecode is seconds, minutes, hours and days. The format is used by several military ranges. Reference IRIG Document 104-59.

2137

The 2137 code is a 25-bit pulse width modulated format used to amplitude modulate a 1000 kHz sine wave carrier. The information contained in the timecode is seconds, minutes and hours. The format is used by certain security organizations.

IEEE-1344 Bit Definition

Bit Position	Bit Definition	Explanation
P50	Year, BCD1	Unit years
P51	Year, BCD2	
P52	Year, BCD4	
P53	Year, BCD8	
P54	Not used	
P55	Year, BCD10	Tens years
P56	Year, BCD20	
P57	Year, BCD40	
P58	Year, BCD80	
P59	P6	Position identifier
P60	Leap second pending	Set to one, 59 seconds prior to leap insertion
P61	Leap second	0 = add second, 1 = delete second
P62	Daylight Savings Time pending	Set to one, 1 second prior to DST change
P63	Daylight Savings Time	1 = DST active
P64	Local offset sign	0 = +, 1 = -
P65	Local offset binary 1	Local offset from UTC time
P66	Local offset binary 2	
P67	Local offset binary 4	
P68	Local offset binary 8	
P69	P7	Position identifier
P70	Local offset half hour bit	0 = none, 1 = half hour time offset added
P71	Time quality binary 1	Time quality indicates clock precision.*
P72	Time quality binary 2	
P73	Time quality binary 4	
P74	Time quality binary 8	
P75	Parity	Odd parity for all preceding data bits
P76-P78	Not used	
P79	P8	Position identifier

* Refer to Appendix A - Time Figure-of-Merit for detailed information. Briefly:

4	time error is < 1 us
5	time error is < 10 us
6	time error is < 100 us
7	time error is < 1 ms
8	time error is < 10 ms
9	time error is > 10 ms, unsynchronized state if never locked to GPS.

Appendix F

IPv6 Information

EndRun Technologies understands that IPv6 is still in the experimental stage with essentially no mainstream deployment. Customers who are not interested in IPv6 need not burden your system with it. You have a choice of an IPv4-only kernel (recommended) or the IPv4/IPv6-kernel. You may freely change this at any time with an easy software download from our website.

To determine which kernel resides in your Tycho GPS check the firmware version using the front-panel keypad/display. Or you can use the console port command **`cat /proc/version`**.

An IPv4-only kernel will have a part number and version similar to:

    6010-0041-000 ver 2.4.26-1

An IPv4/IPv6 kernel will have a part number and version similar to:

    6010-0041-100 ver 2.4.31-IPv6

If you want to change your kernel please refer to ***Appendix B - Upgrading The Firmware*** for instructions. The following text refers to products with the IPv4/IPv6 kernel.

Enabling New IPv6 Capabilities

The presence of an IPv6-capable kernel will automatically enable most of the new IPv6 capabilities. By default, autoconfiguration of the ethernet interface via IPv6 Router Advertisements is enabled. To disable acceptance of Router Advertisements, or to configure a static IPv6 address and default IPv6 gateway, you must either run the interactive **`netconfig`** script or use the front-panel keypad/display. Either method will allow you to configure your ethernet interface for both IPv4 and IPv6 operation. Using the **`netconfig`** script has the advantage that you can also configure the hostname and domainname for the unit, and any nameservers you may want it to have access to.

OpenSSH

By default, **`sshd`** is factory-configured to listen on both IPv4 and IPv6 addresses. It may be forced to listen on either IPv4 only, or IPv6 only by editing the */etc/rc.d/rc.inet2* startup script, where **`sshd`** is started, and then copying it to */boot/etc/rc.d*.

Net-SNMP

By default, **`snmpd`** is factory configured to listen on both IPv4 and IPv6 addresses. This may be changed by editing */etc/rc.d/rc.local* and modifying the agent address argument passed to **`snmpd`** at start-up, and then copying it to */boot/etc/rc.d*.

IPv6-Capable syslog-ng

To enable remote syslogging to an IPv6 host, you will need to edit the new */etc/syslog-ng.conf* file and copy it to */boot/etc*. At boot time, the presence of both the **`syslog-ng`** daemon and the *boot/etc/syslog-ng.conf* file will cause the new IPv6-capable **`syslog-ng`** daemon to be started instead of the previous **`syslogd/klogd`** pair of daemons. These two files remain on the system for backward compatibility with customers' existing */etc/syslog.conf* setups, but they are not IPv6 capable. If you are not currently directing your system logs to a remote host, or you are not using IPv6, then there is little need or benefit to changing to **`syslog-ng`**.

IPv4-Only Protocols

There are several protocols which are not IPv6 capable: **`telnet`** (client and server), **`ftp`** and **`dhcpcd`**. Due to their intrinsic insecurity, **`telnet`** and **`ftp`** are repidly being deprecated, and probably have little business running over an IPv6 network. The address autoconfiguration capabilities of IPv6 make the DHCP protocol less important, however it is likely that the new **`dhcpv6`** capability will appear in a future upgrade.

Appendix *G*

Third-Party Software

The Tycho is running a software product created and/or maintained by open source projects. Open source software comes with its own license. This is printed out for your information below.

The license for the GNU software project requires that we provide you with a copy of all source code covered under the GNU Public License (GPL) at your request. Please contact us with your request and we will mail it to you on a CD. We will charge you a fee for our incurred expenses as allowed for in the license.

GNU General Public License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989,1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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 Lesser 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.

GNU GENERAL PUBLIC LICENSE
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.

END OF TERMS AND CONDITIONS

Appendix *H*

Specifications

GPS Receiver:

L1 Band – 1575.42 MHz
8 Channels, C/A Code

Antenna:

TNC jack on rear panel, Z_{in} = 50Ω
Integral +35 dB gain LNA with bandpass filter for out-of-band interference rejection.
Rugged, all-weather housing capable of operation over –40°C to +85°C temperature extremes.
Mounting via 18” long, ¾” PVC pipe with stainless steel clamps.
50’ low-loss RG-59 downlead cable standard.
Extension cables and low noise pre-amplifiers are available as options.

Local Oscillator:

TCXO is standard ($2.5x10^{-6}$ over -20˚ to 70˚ C).
Medium-Stability OCXO (MS-OCXO) ($4x10^{-9}$ over 0˚ to 70˚ C) is option.
High-Stability OCXO (HS-OCXO) ($1x10^{-9}$ over 0˚ to 70˚ C) is option.

Time to Lock:

< 5 minutes, typical (TCXO).
< 10 minutes, typical (OCXO).

Network I/O:

Rear panel RJ-45 jack
AMD PC-Net Fast III 10/100Base-T ethernet

System Status Indicator:

Sync LED: Green LED pulses to indicate GPS lock status.
Network LED: Amber LED indicates network activity.
Alarm LED: Red LED indicates a serious fault condition.

1 PPS Output:

To change the pulse width refer to the **`cpuopts`** and **`cpuoptsconfig`** ommands in ***Chapter 3 - Control and Status Commands***.
Signal: Positive TTL pulse into 50Ω.
User-Selectable Width: 20 us, 1 ms, 100 ms, 500 ms.
User Calibration: +/- 500 us, 1 ns resolution.
Accuracy: <20 nanoseconds RMS to GPS Time when locked.*
<10 nanoseconds RMS to GPS Time when locked with 10-Nanosecond Calibration Option.
Stability: TDEV < 10 ns, $\tau < 10^5$ seconds, $\sigma_y(\tau) < 1x10^{-13}$ @ $\tau=10^5$ secs.
Connector: Rear-panel BNC labeled “1PPS”.

* <100 nanoseconds to UTC. Constraints in the official GPS specification prohibit claiming an accuracy to UTC better than 100 nanoseconds.

Time Code Output:

To change the time code format refer to the **`cpuopts`** and **`cpuoptsconfig`** commands in ***Chapter 3 - Control and Status Commands.*** Also see ***Appendix E - Timecode Formats***.
Signal: Amplitude-modulated (AM), 3:1 ratio, 1 kHz carrier.
Drive: 1 Vrms into 50Ω.
User-Selectable Formats: IRIG-B120 (IEEE-1344), IRIG-B122, IRIG-B123, NASA36 or 2137.
Connector: Rear-panel BNC labeled “AM CODE”.

Maintenance Console:

See ***RS-232 Serial I/O Port Signal Definitions*** in ***Chapter 3*** for more information.
RS-232 serial I/O on rear panel DB9M jack for secure, local terminal access.
Parameters fixed on 19200 baud, 8 data bits, no parity, 1 stop bit.

Supported IPv4 Network Protocols:

SSH server with “secure copy” utility, SCP
SNMP v1, v2c, v3 with Enterprise MIB
TIME and DAYTIME server
TELNET client/server
FTP client
DHCP client
SYSLOG

Supported IPv6 Network Protocols:

SSH server with “secure copy” utility, SCP
SNMP v1, v2c, v3 with Enterprise MIB
TIME and DAYTIME server
SYSLOG

Power:

90-264 VAC, 47-63 Hz, 0.5 A Max. @ 120 VAC, 0.25 A Max. @ 240 VAC
110-370 VDC, 0.5A Max @ 120 VDC
3-Pin IEC 320 on rear panel, 2 meter line cord is included.

DC Power (option):

38-72 Vdc, 1.5A maximum.
3-position terminal block on rear panel: +DC IN, SAFETY GROUND, -DC IN
(Floating power input: Either “+” or “-” can be connected to earth ground.)

Size:

Chassis:	1.75”H x 17.0”W x 10.75”D
Antenna:	3.5” Dia. x 2.5” H
Weight:	< 5 lb. (2.70 kg.)

Environmental:

Operating Temperature:	0° to +50°C
Operating Humidity:	0 to 95%, non-condensing
Storage Temperature:	-40° to +85°C
Antenna Operating Temperature:	-40° to +85°C

Optional Alarm Output (on CPU Module):

See ***Chapter 4 - Rear-Panel Output Options*** for more information.
Alarm: MMBT2222A open collector, grounded emitter. High impedance in alarm state.
Voltage: 40 VDC, maximum.
Saturation Current: 100 mA, maximum.
Connector: Rear-panel BNC jack or terminal strip labeled "ALARM".

Optional Serial Time Output (on CPU Module):

See ***Chapter 4 - Rear-Panel Output Options*** for more information. Also refer to the **`sertimeopts`** and **`sertimeoptsconfig`** commands in ***Chapter 3 - Control and Status Commands***.
Signal: Output only port at RS-232 levels.
Pinout: Pin 3 is Transmit Data. Pin 5 is GND.
Baud Rate: User-selectable to 4800, 9600, 19200 or 57600.
Parity: User-Selectable to Odd, Even or None.
ASCII Formats: User-Selectable to Sysplex, Truetime, EndRun, EndRunX or NENA.
Connector: Rear-panel DB-9M connector labeled: "SYSPLEX" or "SERIAL TIME".
Signal: RS-232 levels.
Connector: Rear-panel DB-9M connector labeled "Serial Time".
Pinout: Pin 3 is Transmit Data. Pin 5 is GND.

Optional 1 PPS (RS-422) Output (on CPU Module):

See ***Chapter 4 - Rear-Panel Output Options*** for more information. Also refer to the **`cpuopts`** and **`cpuoptsconfig`** commands in ***Chapter 3 - Control and Status Commands***.
Signal: RS-422 levels.
User-Selectable Width: 20 us, 1 ms, 100 ms, 500 ms.
User Calibration: +/- 500 us, 1 ns resolution.
Accuracy: <20 nanoseconds RMS to GPS Time when locked.*
<10 nanoseconds RMS to GPS Time when locked with 10-Nanosecond Calibration Option.
Stability: TDEV < 10 ns, $\tau < 10^5$ seconds, $\sigma_y(\tau) < 1x10^{-13}$ @ $\tau = 10^5$ secs.
Connector: Rear-panel DB-9M jack labeled "1PPS (RS-422)".
Pinout: Pin 3 is +signal. Pin 6 is -signal. Pin 5 is GND.

Optional Programmable TTL Pulse Rate Output (on CPU Module):

See ***Chapter 4 - Rear-Panel Output Options*** for more information. Also refer to the **`cpuopts`** and **`cpuoptsconfig`** commands in ***Chapter 3 - Control and Status Commands***.
Signal: Positive TTL pulse @ 50Ω.
Rate: User selectable to 1, 10, 100, 1K, 10K, 100K, 1M, 5M, 10M PPS and Timecode.
Accuracy: $< 10^{-13}$ to UTC for 24-hour averaging times when locked.
Alignment: Within 10 ns of the other TTL outputs in this unit (except the optional DDS).
Stability: See Stability (Allan Deviation) Table below.
Synthesized Rate: 1 PPS to 10 MPPS in 1 PPS steps with optional DDS Upgrade.
Connector: Rear-panel BNC jack labeled "PROG TTL".

Optional Fixed Pulse Rate Output (on CPU Module):

See ***Chapter 4 - Rear-Panel Output Options*** for more information.
Signal: Positive TTL pulse @ 50Ω.
Rate: Preset at factory and cannot be changed.
Accuracy: $< 10^{-13}$ to UTC for 24-hour averaging times when locked.
Stability: See Stability (Allan Deviation) Table below.
Connector: Rear-panel BNC jack labeled appropriately, i.e. "10 MPPS".

Optional Programmable Digital Buffer Module Outputs:

See ***Chapter 6 - Optional Rear-Panel Outputs*** for more information. Also refer to the **`pluginopts`** and **`pluginoptsconfig`** commands in ***Chapter 3 - Control and Status Commands***.

Quantity: Four outputs.

On-Time Pulse Rates:

Drive: TTL into 50Ω.

Duty Cycle: 50%.

Rate: User selectable to 1, 10, 100, 1K, 10K, 100K, 1M, 5M, 10M PPS.

Alignment: Within 10 ns of the other TTL outputs in this unit (except the optional DDS).

Stability: See Stability (Allan Deviation) Table below.

Digital Timecode:

Drive: TTL into 50Ω (DC level shift).

Format: User-selectable to IRIG-B (000/IEEE-1344, 002, 003), NASA-36, or 2137.

Alignment: Within 10 ns of the other TTL outputs in this unit (except the optional DDS).

Synthesized Rate: 1 PPS to 10 MPPS in 1 PPS steps with optional DDS Upgrade.

Connector: Rear-panel BNC jack.

Optional Analog Timecode Buffer Module Outputs:

See ***Chapter 6 - Optional Rear-Panel Outputs*** for more information. Also refer to the **`cpuopts`** and **`cpuoptsconfig`** commands in ***Chapter 3 - Control and Status Commands*** and ***Appendix E - Timecode Formats***.

Quantity: Four outputs.

Signal: Amplitude-modulated (AM), 3:1 ratio, 1 kHz carrier.

Drive: 1 Vrms into 50Ω.

User-Selectable Formats: IRIG-B120 (IEEE-1344), IRIG-B122, IRIG-B123, NASA36 or 2137.

Connector: Rear-panel BNC jack.

Optional Low Phase Noise Outputs:

See ***Chapter 4 - Rear-Panel Output Options*** for more information. Also refer to the **`pluginopts`** command in ***Chapter 3 - Control and Status Commands***.

Quantity: 4 or 8.

Output Frequency: 5 MHz or 10 MHz.

Output Level: +13 dBm, +/- 2 dBm at 50Ω.

Harmonics: < -45 dBc at 50Ω.

Stability: See Stability (Allan Deviation) Table below.

Channel-to-Channel Isolation: > +75 dB

Connector: Rear-panel BNC jack.

Phase Noise dBc/Hz @ 10 MHz:

Hz	*TCXO*	*Spurs (TCXO)*	*MS-OCXO*	*Spurs (OCXO)*
1	-70	-100	-95	
10	-100	-100	-120	-120
100	-125	-100	-135	-115
1 k	-135	-100	-145	-125
10 k	-140	-100	-145	-125
100 k	-145	-120	-145	-110

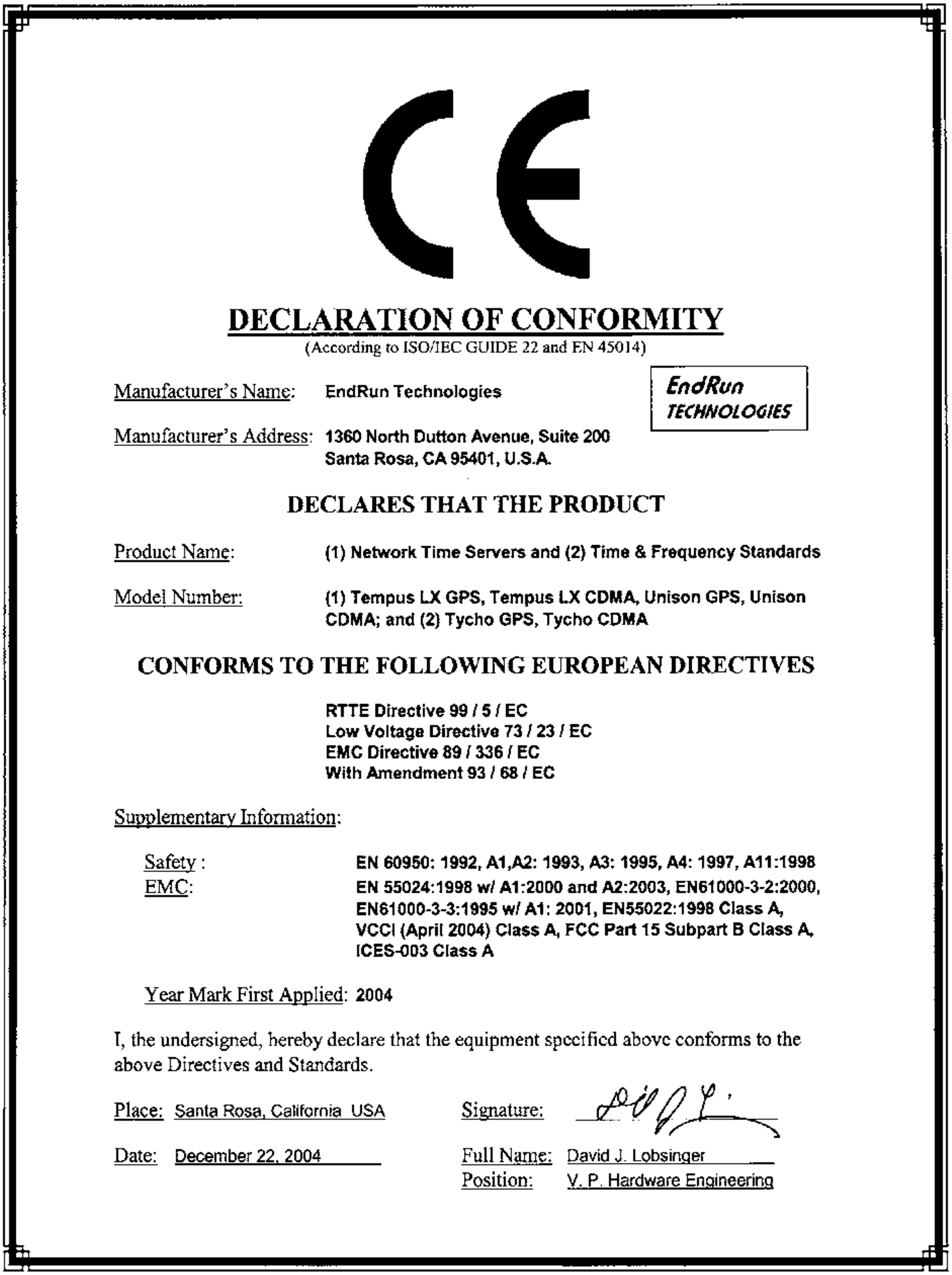
Stability (Allan Deviation) Table:

Tau in Seconds	*TCXO*	*MS-OCXO*
1	$1x10^{-9}$	$7x10^{-12}$
10	$5x10^{-10}$	$9x10^{-12}$
100	$1x10^{-10}$	$1.8x10^{-11}$
1000	$1x10^{-11}$	$1x10^{-11}$
10000	$1x10^{-12}$	$1x10^{-12}$
100000	$1x10^{-13}$	$1x10^{-13}$

CE/FCC Compliance: RTTE Directive 99/5/EC
Low Voltage Directive 73/23/EC
EMC Directive 89/336/EC
With Amendment 93/68/EC

Supplementary Compliance Data:

Safety:	EN 60950;1992, A1,A2: 1993, A3: 1995, A4: 1997, A11:1998
EMC:	EN 55024:1998 w/ A1:2000 and A2:2003, EN61000-3-2:2000, EN61000-3-3:1995 w/ A1:2001, EN55022:1998 Class A, VCCI (April 2004) Class A, FCC Part 15 Subpart B Class A, ICES-003 Class A



DECLARATION OF CONFORMITY

(According to ISO/IEC GUIDE 22 and EN 45014)

Manufacturer's Name: EndRun Technologies

EndRun TECHNOLOGIES

Manufacturer's Address: 1360 North Dutton Avenue, Suite 200
Santa Rosa, CA 95401, U.S.A.

DECLARES THAT THE PRODUCT

Product Name: (1) Network Time Servers and (2) Time & Frequency Standards

Model Number: (1) Tempus LX GPS, Tempus LX CDMA, Unison GPS, Unison CDMA; and (2) Tycho GPS, Tycho CDMA

CONFORMS TO THE FOLLOWING EUROPEAN DIRECTIVES

RTTE Directive 99 / 5 / EC
Low Voltage Directive 73 / 23 / EC
EMC Directive 89 / 336 / EC
With Amendment 93 / 68 / EC

Supplementary Information:

Safety : EN 60950: 1992, A1,A2: 1993, A3: 1995, A4: 1997, A11:1998

EMC: EN 55024:1998 w/ A1:2000 and A2:2003, EN61000-3-2:2000, EN61000-3-3:1995 w/ A1: 2001, EN55022:1998 Class A, VCCI (April 2004) Class A, FCC Part 15 Subpart B Class A, ICES-003 Class A

Year Mark First Applied: 2004

I, the undersigned, hereby declare that the equipment specified above conforms to the above Directives and Standards.

Place: Santa Rosa, California USA

Signature:

Date: December 22, 2004

Full Name: David J. Lobsinger

Position: V. P. Hardware Engineering

Appendix*I*

Software Release Notes for Previous Tycho Users

Version 2.60 of the Linux root file system (RFS) and the new IPv6-capable Linux 2.4.31-IPV6 kernel are now shipping in new products. Both files are also available for download so that you can upgrade your installed units in the field. This is a major upgrade, and features updated versions of all applications, utilities and shared libraries typically installed in an embedded Linux-based system. In addition, the critical open source protocol implementations, OpenSSH, Net-SNMP and Syslog-ng are now IPv6-capable, along with other various support daemons and configuration utilities that need to understand IPv6 addresses.

Easy Field-Installable Upgrade

The new system detects the presence of an IPv6-capable kernel and enables the IPv6 configuration menus and command line utilities automatically. As with all of our firmware upgrades, we have designed the upgrade to be seamless for existing customers, which means that after applying the update, your existing configuration settings and passwords will continue to function properly.

Freedom of Choice

EndRun Technologies understands that IPv6 is still in the experimental stage with essentially no mainstream deployment. Customers who are not interested in IPv6 need not perform the Linux 2.4.31-IPV6 kernel upgrade procedure, and your systems will continue to behave as before. Customers buying new products may choose to have the IPv6-capable kernel installed at the factory. The default will be the previous Linux 2.4.26 IPv4-only kernel.

Performing the Upgrade

Performing the 2.60 RFS upgrade is identical to the current procedure (see your User Manual, Appendix B, Performing the Linux/NTP Upgrade), and must be performed first if you are also planning to upgrade your kernel. The IPv6 Linux 2.4.31 kernel upgrade procedure is new, and a new utility, **`upgradekernel`** has been added to the 2.60 RFS to facilitate and failsafe this procedure. First you need to upload the new compressed kernel image file to a temporary location on the file system, using **`scp`**. (Alternatively, you could ftp from your timeserver to an ftp server on your network and download the file). Then the kernel upgrade utility is executed with a single argument passed on the command line: the path to the previously uploaded kernel image file. Like this, for example:

```
upgradekernel /tmp/newkernelimage
```

The kernel upgrade utility verifies the integrity of the file, reads the kernel version information, presents it to you and asks you to verify before replacing the old kernel image. If you verify, it will then erase the old image and write the new one in its place. The erase and write operation takes about 10 seconds. ***A power failure during this time would render the unit unbootable, so it is highly advisable to plug the unit into a UPS while performing the upgrade.***

Enabling New IPv6 Capabilities

The presence of an IPv6 capable kernel will automatically enable most of the new IPv6 capabilities. By default, autoconfiguration of the ethernet interface via IPv6 Router Advertisements is enabled. To disable acceptance of Router Advertisements, or to configure a static IPv6 address and default IPv6 gateway, you must either run the interactive **netconfig** script or, if your unit is so equipped, use the front-panel keypad and display. Either method will allow you to configure your ethernet interface for both IPv4 and IPv6 operation. Using the **netconfig** script has the advantage that you can also configure the hostname and domainname for the unit, as well as any nameservers you may want it to have access to.

OpenSSH

Starting with the 2.60 RFS, **sshd** is no longer started by the superserver daemon, inetd. If you have a previously reconfigured */boot/etc/inetd.conf*, the */etc/rc.d/rc.inet2* startup script will detect it and remove the line that allows **sshd** to be started by **inetd**. By default, **sshd** is factory configured to listen on both IPv4 and IPv6 addresses. It may be forced to listen on either IPv4 only, or IPv6 only by editing the */etc/rc.d/rc.inet2* startup script, where **sshd** is started, and then copying it to */boot/etc/rc.d*.

Net-SNMP

By default, **snmpd** is factory configured to listen on both IPv4 and IPv6 addresses. This may be changed by editing */etc/rc.d/rc.local* and modifying the agent address argument passed to **snmpd** at start-up, and then copying it to */boot/etc/rc.d*.

The 2.60 RFS now contains the Net-SNMP open source implementation, which replaces the older UCD-SNMP implementation, which did not support IPv6. There are several new directives in the /etc/snmpd.conf related to IPv6. If you are upgrading and you need IPv6 capability with SNMP, you should merge any changes that you may have made to the previous *snmpd.conf* file (which would be stored in */boot/etc/snmpd.conf*) into the new *snmpd.conf* file, like trapsink addresses and community strings. Using the new *snmpd.conf*, you can set up any IPv6 trapsink addresses. If you are using snmpv3 secure access, you will need to perform the **createUser** operations to the new */boot/net-snmp/snmpd.conf* persistent configuration file. The older */boot/ucd-snmp* directory is no longer used for this.

New IPv6-Capable syslog-ng

To enable remote syslogging to an IPv6 host, you will need to edit the new */etc/syslog-ng.conf* file and copy it to */boot/etc*. At boot time, the presence of both the **syslog-ng** daemon and the *boot/etc/syslog-ng.conf* file will cause the new IPv6-capable **syslog-ng** daemon to be started instead of the previous **syslogd/klogd** pair of daemons. These two files remain on the system for backward compatibility with customers' existing */etc/syslog.conf* setups, but they are not IPv6 capable. If you are not currently directing your system logs to a remote host, or you are not using IPv6, then there is little need or benefit to changing to **syslog-ng**.

Remaining IPv4-Only Protocols

There remain several protocols in the 2.60 RFS which are not IPv6 capable: **`telnet`** (client and server), **`ftp`** and **`dhcpcd`**. Due to their intrinsic insecurity, **`telnet`** and **`ftp`** are rapidly being deprecated, and probably have little business running over an IPv6 network. The address autoconfiguration capabilities of IPv6 make the DHCP protocol less important, however it is likely that the new **`dhcpv6`** capability will appear in a future upgrade.

EndRun
TECHNOLOGIES
"Smarter Timing Solutions"

1360 N. Dutton Ave. Suite 200
Santa Rosa, CA 95401
TEL 1-877-749-3878
FAX 707-573-8619
www.endruntechnologies.com