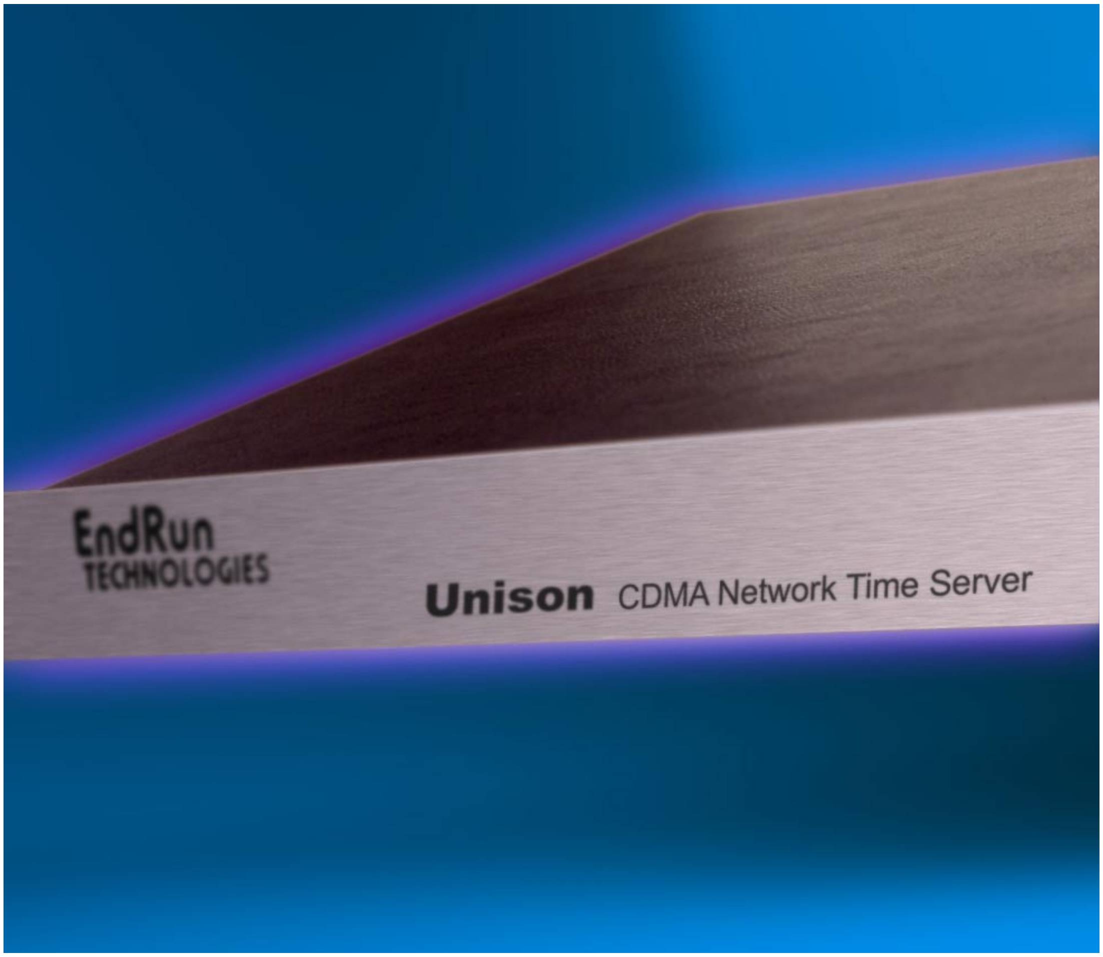
EndRun TECHNOLOGIES
"Smarter Timing Solutions"

Unison *CDMA*

Network Time Server



User Manual

Unison *CDMA*

Network Time Server User Manual

Preface

Thank you for purchasing the Unison Network Time Server. Our goal in developing this product is to bring precise, Universal Coordinated Time (UTC) into your network quickly, easily and reliably. Your new Unison 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.

Founded in 1998 and headquartered in Santa Rosa, California, EndRun Technologies is the undisputed leader in the time and frequency distribution technology based on the Code Division Multiple Access (CDMA) mobile telecommunications infrastructure. With innovative designs and painstaking attention to the details of efficient manufacturability, EndRun is the first to bring this technology to the broad synchronization market at prices small businesses can afford.

The instruments produced by EndRun Technologies have been selected as the timing reference for such rigorous applications as computer synchronization, research institutions, aerospace, network quality-of-service monitoring, satellite earth stations, and calibration laboratories.

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. USM3016-0000-000 Revision 11
September 2009

About This Manual

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

Introduction – The Unison, how it works, where to use it, its main features.
Basic Installation – How to connect, configure and test your Unison with your network.
Client Set-Up – Two sections; one for Unix-like platforms and one for Windows NT/2000/XP.
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 three 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 outside the United States.

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.

Extended Warranty

The MTBF (Mean Time Between Failures) for this product is 225,000 hours (25 years). After the initial warranty period it is most cost-effective for the customer to repair the unit on an “as needed basis”, rather than pay for an extended warranty or the annually recurring fees of a service contract..

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
2270 Northpoint Parkway
Santa Rosa, CA 95407
U.S.A.
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 Unison is a precision server of Universal Coordinated Time (UTC) that can be connected via a 10/100Base-T ethernet port to any TCP/IP network. In its most basic operation, it sends Network Time Protocol (NTP)/Simple Network Time Protocol (SNTP) reply packets in response to NTP/SNTP request packets which it has received from clients. The timestamps it sends in its NTP/SNTP reply packets are accurate to less than one-hundred microseconds. NTP/SNTP client software is available for virtually all operating systems.

The Unison is composed of a Code Division Multiple Access (CDMA) 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 Unison application software is via FLASH memory.

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.*

CDMA Timing-How It Works

The CDMA time and frequency engine in the Unison receives transmissions from base stations, also known as cell sites, that are operating in compliance with the TIA/EIA IS-95 standard for Code Division Multiple Access (CDMA) mobile telecommunications. This system requires a means of synchronizing the base stations throughout the network so that neighboring cells do not interfere with each other and so that calls can be efficiently transferred between the base stations, without interruption, as the mobile user traverses the cell coverage areas. This ‘soft hand-off’ feature means that the mobile telephone must be able to ‘hitlessly’ drop one base station and pick up the next one. To do this, the telephone must be able to calculate the relative difference in time between the codes that modulate the signals from each of the base stations, which again, requires that the base stations be synchronized.

The system designers chose the Global Positioning System (GPS), which is itself a CDMA-based system, as the means of maintaining synchronization, and they defined *system time* to be *GPS time*. Each base station throughout the system contains one or more high-performance GPS timing receivers with sophisticated algorithms that control either an extremely stable ovenized quartz crystal oscillator or a Rubidium vapor atomic frequency standard. Such elaborate means are needed to meet the very difficult operating specifications required by the TIA/EIA IS-95 standard. The base station time synchronization must remain within 10 microseconds of GPS time over periods as long as twenty-four hours during which GPS satellite signals might not be available (typically due to antenna/cable failure, damage or vandalism) and in an environment where large ambient temperature swings may occur. Equipment capable of meeting these requirements is at the current state-of-the-art.

The CDMA time and frequency engine in the Unison receives the same initialization signals transmitted by the base stations that are used by the mobile telephones to establish their synchronization to system time. The mobile telephones cannot communicate in the system until they have established synchronization with the received spread spectrum encoded waveform. Unlike the mobile telephones, once this synchronization has occurred, the CDMA time and frequency engine in the Unison has all of the information that it needs to perform its function of delivering accurate UTC time to a network of computers. The mobile telephone must decode much more information, establish two-way communications with the base station, and be a paid subscriber to performs its function of placing and receiving calls.

All of this means that during normal operation, the quality of the timing information being transmitted from each of the base stations is virtually a repeat of that directly obtainable from the GPS. The big difference is that the received signal strengths from the base stations are a minimum of 30 dB larger than those from the GPS satellites, which is why you can usually talk on your cell phone indoors. Due to the nature of the IS-95 spread spectrum CDMA modulation scheme, this timing information may be extracted by a well-designed receiver with a precision of a few nanoseconds. The CDMA time and frequency engine in the Unison does just that, and for this reason, we call our technology ‘indirect GPS’.

Where to Use It

First, the Unison must be deployed in a *cellular* or *PCS* IS-95 CDMA coverage area. *Cellular* is a commonly used term implying that the frequency band for the base station carrier transmissions is 824-895 MHz. This is in contrast to *PCS*, which implies operation in the 1850-1990 MHz frequency band. If available, the Unison uses the cellular frequency band because it provides much better propagation characteristics in regards to building penetration and maximum receivable range from the transmitter. In regions lacking cellular coverage, the unit can be set to receive the PCS signals. In general, if your CDMA telephone works where you plan to install the Unison, then your Unison will work properly there.

Because the Unison has been designed to operate in conjunction with existing public domain NTP/SNTP client software that has been created for use with similar time servers, it may be used in any computer network environment that is using TCP/IP protocols. Although client software is available for all platforms, for the most precise applications, the Unix-like operating systems are best supported.

Main Features

Performance, Reliability and Economy

The Unison provides high performance and reliability combined with low power consumption and cost. Its internal sub-assemblies are fabricated using state-of-the-art components and processes and are integrated in a solid, high-quality chassis.

Flexibility

It supports a variety of TCP/IP network protocols compatible with a variety of platforms and operating systems.

Easy Installation

Its standard 1U high, 19” rack-mountable chassis and indoor-mounted, magnetic-base antenna make installation simpler compared to *direct* GPS products. The antenna and rack-mount chassis may be mounted in any convenient location. Connect it to your network via the rear panel mounted, 10/100Base-T RJ-45 connector and plug in the AC power cord. Initial network configuration is automatic on networks using the Dynamic Host Configuration Protocol (DHCP). Manual network configuration is via the RS-232 serial I/O port and a simple Linux shell script.

Free FLASH Upgrades

Firmware and configurable hardware parameters are stored in non-volatile FLASH memory, so the Unison 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.

Chapter *Two*

Basic Installation

This chapter will guide you through the most basic checkout and physical installation of your Unison. Subsequent chapters and appendices will give you the information needed to configure your installation for the maximum performance in your operating environment. General NTP client setup instructions will also be supplied to get you started using your Unison quickly.

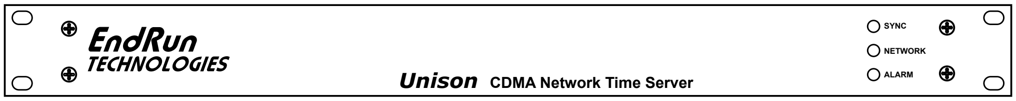
Basic familiarity with TCP/IP networking protocols like **`ping`**, **`telnet`** *and* **`ftp`** *is required. Though some familiarity with Linux or other Unix-like operating systems would be helpful, it is not essential. If you satisfy these conditions, the instructions provided herein should guide you to a successful installation.*

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 Unison shipment typically contains:

- Unison (part # 3016-0001-000 or #3016- variant)
- Unison User Manual (part #USM3016-0000-000)
- IEC 320 AC Power Cord (part #0501-0003-000)
  (This part will not be present if using the DC power option.)
- DB-9F to DB-9F 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)
- Magnetic mount antenna/cable assembly (part #502-0007-001)

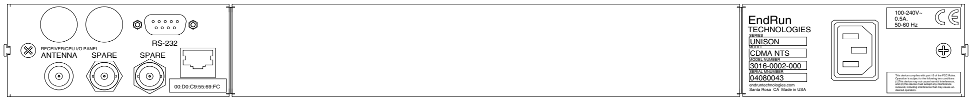
Unison Physical Description



Sync LED
This green LED flashes to indicate synchronization status.

Network LED
This amber LED illuminates when the Unison 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.


RECEIVER/CPU I/O PANEL
ANTENNA
SPARE
SPARE
RS-232
00:D0:C9:55:69:FC
EndRun
TECHNOLOGIES
SERIES
UNISON
MODEL
CDMA NTS
MODEL NUMBER
3016-0002-000
SERIAL MNUMBER
04080043
endruntechnologies.com
Santa Rosa CA Made in USA
100-240V~
0.5A.
50-60 Hz
CE
This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions (1)This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

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 Unison. This console allows the user to initialize and maintain the Unison. See ***Chapter 5 - 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
(Option)
This BNC connector provides the optional 1PPS TTL output. 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 I - Specifications*** for the 1PPS 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 console command **`cpuoptsconfig`**. See pinout details in ***Appendix I - Specifications*** for the 1PPS RS-422 Output.

AM Code Jack
(Option)
This BNC connector provides the optional amplitude-modulated timecode output, and is usually labeled “SPARE”. The timecode output is normally IRIG-B122 when shipped from the factory, but can be changed via command **`cpuoptsconfig`**. See details in ***Appendix I - 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 details in ***Appendix H - Specifications*** for the Alarm Output.

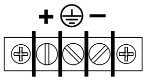
Prog TTL Jack
(Option)
This BNC connector provides the optional Programmable TTL pulse rate output and is usually not installed. If installed, see signal definition in ***Appendix I - Specifications***. This pulse rate is normally shipped from the factory as 10MPPS but can be changed via command **`cpuoptsconfig`**.

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 signal definition in ***Appendix I - Specifications*** for the Fixed Rate 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 ***Appendix G - Serial Time Output.***

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.



Performing an Initial Site Survey

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

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

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

3. Plug the other end into the AC input connector on the chassis rear panel of the Unison.

Place the antenna on a flat, preferably metallic surface while the unit is searching for the signal. Make sure that it is not blocked by large metallic objects closer than one meter. Although the antenna should normally be installed in a vertical orientation, usually multipath conditions due to signal reflections indoors cause at least some of the signal to be horizontally polarized, so do not be surprised if you find that the unit will work with the antenna oriented either way. Multipath conditions can also cause another effect: signal cancellation. Since the wavelength of the signal is only about 12 to 30 centimeters, movement of the antenna just a few centimeters can sometimes cause significant signal strength changes.

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 has detected a CDMA signal, the green Sync Status LED will begin to flash very slowly (about a .4 Hz rate).

4. As the unit locks onto the CDMA signal and begins to decode the timing data, the green Sync Status LED will flash very rapidly (about a 6 Hz rate) until the data is fully decoded.

5. 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 CDMA time and frequency engine has fully synchronized, and you may proceed to permanently mounting the chassis and antenna in the desired location.

If this sequence has not occurred within twenty minutes, you should move the antenna and/or change its orientation and re-try. If you are unable to find an antenna location where the unit will acquire the CDMA signals, you may not have *cellular* coverage in your area or the signal might be too weak in your facility. First, using the **`setcdmachannelset`** command, try changing the channelset on your unit to operate with the PCS frequencies. If you are still unable to receive signals, you should continue to try for at least a day, since base stations are taken down for service from time to time.

If you have a CDMA phone, see if it will work in *digital* mode. If it will, then your Unison may be damaged and should be returned to the factory for repair or exchange.

Installing the Unison

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 Unison

Using standard 19” rack mounting hardware, mount the unit in the previously surveyed location.

CAUTION

Ground the unit properly with the supplied power cord.

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

Do not install the Unison 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 Unison 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.

Installing the Antenna

Make sure that the antenna is not blocked by metallic objects that are closer than about one meter. A good location is the top surface of the equipment rack into which the unit has been installed. Ideally it should be mounted vertically, as the transmitted signals are vertically polarized. When indoors, however, multipath conditions may exist. This means that reflected signals may be present with either vertical or horizontal polarization, so your antenna might work in either orientation. After mounting the unit and antenna, verify that it still acquires and tracks a CDMA signal.

Connecting and Configuring Ethernet

Connect one end of the CAT-5 patch cable supplied with your Unison 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 Unison will attempt to configure the ethernet interface automatically via the Dynamic Host Configuration Protocol (DHCP). The Unison 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 Unison 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 Unison 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 Unison 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 Unison 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 Unison.

To test serial communications with the Unison 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 Unison.

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

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 5 - 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 Unison 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 Unison. 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 Unison 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 CDMA
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:

```
Unison CDMA 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 Network Time Protocol daemon...
Starting the SNMP daemon...
Starting the system logfile manager...
Starting the system watchdog...woof!
```

During this process, the factory default UnisonCDMA_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 Unison login prompt is displayed:

```
********************************************************************************
*           Welcome to Unison CDMA console on:  cntp.your.domain
*           Tue Feb 20  2001 21:47:03 UTC
********************************************************************************

cntp login:
```

Here you may log in as "cntpuser" with password "Praecis" or you may log in as the "root" user with password "endrun_1". When logged in as "cntpuser", 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 Unison CDMA and shows the software part number, version and build date:

```
Unison CDMA 6010-0042-000 v 1.00 Wed May  9 14:17:44 UTC 2002
Unison CDMA (root@cntp:~)->
```

This last line is the standard Unison CDMA shell prompt. The Unison uses the **`bash`** shell, which is the Linux standard, full-featured shell. After configuring the unit, you should change the passwords using the **`cntppasswd`** 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 5 - RS-232 Serial I/O Port Signal Definitions*** for the signal connections for the Unison.

NOTE

You must use a null-modem cable or adapter if you are connecting the Unison 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 Unison, you may procede to configuring the network parameters. Then you can communicate with the Unison over the network using **`telnet`** or **`ssh`** and synchronize your network computers to UTC using NTP.

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:

```
Unison CDMA(root@cntp)-> netconfig
********************************************************************************
******************** Unison CDMA Network Configuration  ************************
********************************************************************************
*                                                                              *
*   This script will configure the TCP/IP network parameters for your          *
*   Unison CDMA. 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      *
*   Unison CDMA, 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 Unison CDMA. Only the base
hostname is needed, not the domain.
Enter hostname: cntp

---DOMAIN name setting

Set the domain name. Do not supply a leading '.'
Enter domain name for cntp: your.domain

---STATIC IP ADDRESS setting

Set the IP address for the Unison CDMA. Example: 111.112.113.114
Enter IP address for cntp (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 Unison CDMA's current IP address, full hostname, and base hostname:
192.168.1.245       cntp.your.domain    cntp

---DOMAIN NAMESERVER(S) address setting

Will your Unison CDMA 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 Unison CDMA 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 Unison CDMA 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 Unison and re-boot it. There are two ways to do this:

1. Cycle power to the Unison.

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

```
Unison CDMA(root@cntp:~)-> shutdown -r now
```

If you are using the RS-232 serial I/O port to communicate with the Unison, 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 Unison 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 Unison 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`**:

```
Unison CDMA(root@cntp:~)-> 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**:

```
Unison CDMA(root@cntp:~)-> 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 Unison has been successfully configured to operate on your network and you are ready to check operation of the Unison 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:

```
Unison CDMA(root@cntp:~)-> 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 Unison 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 Unison. Alternatively, you could **ping** one of your servers or workstations from the Unison shell prompt to test the setup.

Once you have successfully established network communications with the Unison, you may perform all maintenance and monitoring activities via **telnet** and **ftp**. The Unison provides both client and server operation using **telnet**. For security reasons as well as to reduce the memory footprint in the Unison, 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 Unison. Both of these protocols are supported in the Unison via the OpenSSH implementations for Linux. Refer to ***Appendix A - Security*** for more information about the secure shell protocol.

Using Telnet

When establishing a **`telnet`** connection with your Unison, 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 Unison, this banner will be displayed:

```
********************************************************************************
*           Welcome to Unison CDMA telnet console on:  cntp.your.domain
********************************************************************************

cntp login:
```

Here you may log in as “cntpuser” with password “Praecis”. When logged in as “cntpuser”, 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 Unison CDMA and shows the software part number, version and build date:

```
Unison CDMA 6010-0004-000 v 1.00 Wed May 16 14:17:44 UTC 2002
Unison CDMA(root@cntp:~)->
```

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

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

```
Unison CDMA(root@cntp:~)-> 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 Unison, logging in directly as *root* is permitted. When you log in as *root* via a **`ssh`** session with the Unison, this banner will be displayed:

```
********************************************************************************
*              Welcome to Unison CDMA SSH console on:  cntp.your.domain
********************************************************************************

root@cntp.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 Unison and shows the software part number, version and build date:

```
Unison CDMA 6010-0042-000 v 1.00 Fri Aug 20 14:17:44 UTC 2004
Unison CDMA(root@cntp:~)->
```

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

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

Configuring the Network Time Protocol

Now that the network has been configured and tested, you may configure the operation of the NTP server. By default, the Unison is configured to respond to NTP requests from clients that may or may not be using MD5 authentication. If the clients are using MD5 authentication, they must be configured properly with the same MD5 authentication keys as the Unison. If you need to modify the factory default Unison MD5 keys (recommended) or set up broadcast/multicast operation, then you will need to re-configure the NTP subsystem. You may perform the configuration from either a **telnet** or **ssh** session, or the local RS-232 console.

NOTE

If you would like to configure your server for multicast operation, configure it as you would for broadcast operation, with the exception that you must enter this specific NTP multicast address: 224.0.1.1, when you are prompted to enter the broadcast address.

Configuring NTP Using the Network Interface or Serial Port

The following is a transcript of the question and answer configuration utility provided by **ntpconfig**. The user entered parameters are underlined:

```
Unison CDMA(root@cntp:~)-> ntpconfig

*******************************************************************************
**********************Network Time Protocol Configuration**********************
*******************************************************************************
*                                                                             *
*   This script will allow you to configure the ntp.conf and ntp.keys files   *
*   that control Unison NTP daemon operation.                                 *
*                                                                             *
*   You will be able to create new MD5 authentication keys which are stored   *
*   in the ntp.keys file.                                                     *
*                                                                             *
*   You will be able to update the authentication related commands in the     *
*   ntp.conf file.                                                            *
*                                                                             *
*   You will be able to configure the “broadcast” mode of operation, with     *
*   or without authentication.  If you supply the multicast address instead   *
*   of your network broadcast address, then you will be able to configure     *
*   the time-to-live of the multicast packets.                                *
*                                                                             *
*   The changes you make now will not take effect until you re-boot the       *
*   Unison CDMA.  If you make a mistake, just re-run ntpconfig prior to       *
*   re-booting.                                                               *
*                                                                             *
*   You will now be prompted for the necessary set up parameters.             *
*                                                                             *
```

```
*******************************************************************************
*******************************************************************************

---MD5 Keyfile Configuration

Would you like to create a new ntp.keys file? ([y]es, [n]o) y

You will be prompted for a key number (1 - 65534), then the actual key.
When you have entered all of the keys that you need, enter zero at the next
prompt for a key number.

MD5 keys may contain from 1 to 31 ASCII characters.  They may not contain
SPACE, TAB, LF, NULL, or # characters!

Enter a key number (1-65534) or 0 to quit: 1

Enter the key (1-31 ASCII characters): EndRun_Technologies_LLC

Writing key number: 1 and Key: EndRun_Technologies_LLC to ntp.keys

Enter a key number (1-65534) or 0 to quit: 2

Enter the key (1-31 ASCII characters): Tempus_CDMA

Writing key number: 2 and Key: Tempus_CDMA to ntp.keys
Enter a key number (1-65534) or 0 to quit: 0
---NTP Authentication Configuration
Do you want authentication enabled using some or all of the keys in
the ntp.keys file? ([y]es, [n]o) y

You will be prompted for key numbers (1 - 65534), that you want NTP to
“trust”.  The key numbers you enter must exist in your ntp.keys file.  If you
do not want to use some of the keys in your ntp.keys file, do not enter them
here.  NTP will treat those keys as “untrusted”.

Clients that use any of the “trusted” keys in their NTP polling packets will
receive authenticated replies from the Unison CDMA.  When you have entered
all of the “trusted keys” that you need, enter zero at the next prompt for a
key number.

Enter a trusted key number (1-65534) or 0 to quit: 1

Enter a trusted key number (1-65534) or 0 to quit: 2

Enter a trusted key number (1-65534) or 0 to quit: 0

---NTP Broadcast/Multicast Configuration

Would you like to enable broadcast/multicast server operation? ([y]es, [n]o) y

Set the network broadcast/multicast address for the Unison CDMA to use.  For
broadcast mode, this address is the all 1’s address on the sub-net.
Example: 111.112.113.255
For multicast operation, it is this specific address:  224.0.1.1

Enter IP address for NTP broadcast/multicast operation (aaa.bbb.ccc.ddd): 224.0.1.1

You have selected multicast operation.  Enter the number of hops that
are needed for the multicast packets on your network (positive integer): 1

It is highly recommended that authentication be used if you are using NTP in
broadcast/multicast mode.  Otherwise clients may easily be “spoofed” by a fake NTP
server.  You can specify an MD5 key number that the Unison CDMA will use in its
```

```
broadcast/multicast packets.  The clients on your network must be configured to use
the same key.

Would you like to specify an MD5 key number to use with
broadcast mode? ([y]es, [n]o) y

Enter the MD5 key number to use (1-65534): 2

*******************************************************************************
*******************************************************************************
*                                                                             *
*    The Unison CDMA Network Time Protocol configuration has been updated.    *
*                                                                             *
*              Please re-boot now for the changes to take effect.             *
*                                                                             *
*******************************************************************************
*******************************************************************************
```

Configuring the Unison as a Stratum 2 Server

Operating the Unison as a Stratum 1 Server is the recommended mode. You may operate the unit as a Stratum 2 server but since there are innumerable ways to configure your network with Stratum 2 servers, specific insructions for how to do that are beyond the scope of this manual. General instructions are:

Edit NTP.CONF

You must edit the ntp.conf file in order to point your Stratum 2 server at a Stratum 1 server. Edit /etc/ntp.conf and add your server line(s). Here is an example:

```
  server 192.168.1.1
```

IMPORTANT!! Do not remove the server lines for the refclock. Even if your Time Server is not connected to an antenna - the refclock server lines must remain.

Now save the edited file and copy it to the non-volatile flash partition with this command:

```
  cp -p /etc/ntp.conf /boot/etc
```

Mask Alarm

In Stratum 1 operation an alarm will be indicated when there is a loss of signal. For Stratum 2 operation you may not want to see this alarm. You can mask it (prevent it from showing) by using the console port (serial/network) command **`setsigfltmask`**.

Chapter Three

Setting Up NTP Clients on Unix-like Platforms

To configure your Unix-like computer to use your Unison, you must have successfully completed the Basic Installation procedures in Chapter 2. This manual is not a 'How-To' on installing and using NTP; basic approaches to NTP client configuration for operation with the Unison will be described. It is expected that you are, or have access to, a capable Unix/Linux system administrator and know more than a little about installing distributions from source code. Installation must be performed by a user with root priviledges on the system. If you have never used NTP, then you should spend some time reading the on-line documents, especially the Distribution Notes, FAQ and Configuration subject matter, which are available at: http://www.ntp.org

Although all the information is available at the above site, the following are excellent tutorials on setting up NTP and are easier to understand:

http://www.sun.com/solutions/blueprints/0701/NTP.pdf
http://www.sun.com/solutions/blueprints/0801/NTPpt2.pdf
http://www.sun.com/solutions/blueprints/0901/NTPpt3.pdf

If you have a new server, many problems may be solved by the helpful people who participate in the Internet news group devoted to NTP at: comp.protocols.time.ntp.

Three methods of using the Unison with NTP clients on Unix-like platforms will be described:

Basic: This is the simplest, and will operate without MD5 authentication. **NTP beginners should always perform this setup first**.

MD5: This method is trickier only because MD5 keys must be set up and distributed accurately to the NTP clients in a secure way. The Unison is factory configured to authenticate its replies to NTP MD5 clients using its default set of keys.

Broadcast/Multicast: This method simplifies configuration of the clients on large networks since specific server addresses need not be configured in each client's */etc/ntp.conf* file. It can be configured either with or without MD5 authentication. However, it is highly recommended that authentication be configured when using broadcast/multicast mode due to the relative ease with which a fake NTP server can take over the clock setting of the broadcast/multicast clients on the network.

Basic NTP Client Setup

Basic setup is relatively simple, if:

- You have been able to successfully communicate with the Unison on your network.

- You have installed NTP on your client computer.

Configure NTP

You must edit the *ntp.conf* file which **`ntpd`**, the NTP daemon, looks for by default in the the */etc* directory. Add this line to the ntp.conf file:

```
server 192.168.1.245
```

This line tells **`ntpd`** to use the NTP server at address 192.168.1.245 in addition to any other servers which might also be configured in the client's *ntp.conf* file.

Re-start **`ntpd`** to have it begin using the Unison server. Use the NTP utility **`ntpq`** to check that **`ntpd`** is able to communicate with the Unison. After issuing the command

```
ntpq
```

you will see the **`ntpq`** command prompt:

```
ntpq>
```

Use the command

```
peers
```

to display the NTP peers which your computer is using. One of them should be the Unison server which you have just configured. You should verify that it is being 'reached'. (You may have to continue issuing the peers command for a minute or two before you will see the 'reach' count increment.) If you have other peers configured, verify that the offset information for the Unison server peer and your other peers is in agreement to within a few milliseconds, assuming that the other peers are synchronized to that level of accuracy.

It may also be useful to start the NTP daemon in 'debug' mode (**`ntpd   -d`**) to confirm successful configuration. Refer to the NTP documentation for detailed usage of these debug utilities.

MD5 Authenticated NTP Client Setup

MD5 authenticated setup is relatively simple, if:

- You have been able to successfully communicate with the Unison on your network.

- Your Unison has been configured to perform authentication either by factory default, or by running the **`ntpconfig`** shell script. The example Unison authentication configuration shown in ***Chapter 2 - Configuring the Network Time Protocol*** will be assumed in the example configuration commands shown here.

- You have installed NTP on your client computer.

- You have successfully performed the ***Basic NTP Client Setup*** on your client computer.

Create the ntp.keys File

You must create a file named *ntp.keys* in the */etc* directory. It must be a copy of the one residing in the */etc* directory of your Unison. You can **`telnet`** into your Unison and start an **`ftp`** session with your client computer to send the Unison's */etc/ntp.keys* file to your client computer, use the secure copy utility **`scp`**, or you can just use a text editor on your client computer to create an equivalent file.

IMPORTANT

Handling of the ***/etc/ntp.keys*** file is the weak link in the MD5 authentication scheme. It is very important that it is owned by ***root*** and not readable by anyone other than ***root***.

After transferring the file by **`ftp`**, and placing it in the */etc* directory on the client computer, issue these two commands at the shell prompt:

```
chown root.root /etc/ntp.keys
chmod 600 /etc/ntp.keys
```

Configure NTP

You must edit the *ntp.conf* file which **`ntpd`**, the NTP daemon, looks for by default in the */etc* directory. Assuming that you have created two trusted keys as shown in the example in the previous chapter, add these lines to the end of the *ntp.conf* file:

```
keys /etc/ntp.keys
trustedkey 1 2
```

Modify the line added previously in ***Basic NTP Client Setup*** so that authentication will be used with the Unison server using one of the trusted keys, in this case key # 1:

```
server 192.168.1.245 key 1
```

Re-start **`ntpd`** to have it begin using the Unison server with MD5 authentication. Use the NTP utility **`ntpq`** to check that **`ntpd`** is able to communicate with the Unison. After issuing the command

```
ntpq
```

you will see the **`ntpq`** command prompt:

```
ntpq>
```

Use the command

```
peers
```

to display the NTP peers which your computer is using. One of them should be the Unison server which you have just configured. You should verify that it is being 'reached'. (You may have to continue issuing the peers command for a minute or two before you will see the 'reach' count increment.)

You can verify that authentication is being used by issuing the command

```
associations
```

to display the characteristics of the client server associations. In the “auth” column of the display, you should see “OK” for the row corresponding to the Unison server. If you see “bad”, you should wait a few minutes to be sure that there is a problem since “bad” is the initial state of this setting. If the “bad” indication persists then you must check your configuration for errors. Typically this is due to a typing error in creating the */etc/ntp.keys* file on the client that causes a mismatch between the keys being used by the server and client. (If you transfer the file by **`ftp`** or **`scp`**, this shouldn’t be a problem.) It is also possible to have a typing error in the */etc/ntp.conf* file that causes the needed key to not be included in the “trustedkey” list.

Broadcast/Multicast NTP Client Setup

Broadcast/multicast client setup is relatively simple, if:

- You have been able to successfully communicate with the Unison on your network.

- Your Unison has been configured to perform broadcasts or multicasts by running the **`ntpconfig`** shell script. (This is not the factory default configuration, so be sure to run **`ntpconfig`**.) If you are going to use MD5 authentication, your Unison must have been configured to operate with authentication in the broadcast/multicast mode, and you must know which of the trusted keys it is using for broadcast/multicast operation. The example Unison configuration shown in ***Chapter 2 - Configuring the Network Time Protocol*** will be assumed in the example configuration commands shown here.

- You have installed NTP on your client computer.

- You have successfully performed the ***MD5 Authenticated NTP Client Setup*** on your client computer, if you plan to use MD5 authentication.

Configure NTP

You must edit the *ntp.conf* file which **`ntpd`**, the NTP daemon, looks for by default in the the */etc* directory. Assuming that your Unison server has been configured to use key 2 for broadcast authentication as shown in the example in chapter 2, make sure that key 2 is included in the **`trustedkey`** line, and add this line to the end of the *ntp.conf* file:

```
broadcastclient
```

If you are not using MD5 authentication, you would add these lines:

```
disable auth
broadcastclient
```

If you are using multicast instead of broadcast mode, you would replace the **`broadcastclient`** keyword with the **`multicastclient`** keyword. You may remove the line added previously in ***Basic NTP Client Setup***:

```
server 192.168.1.245
```

or the authenticated version added in ***MD5 Authenticated NTP Client Setup***:

```
server 192.168.1.245 key 1
```

Re-start **`ntpd`** to have it begin using the Unison as a broadcast or multicast server. Use the NTP utility **`ntpq`** to check that **`ntpd`** is able to communicate with the Unison. After issuing the command

```
ntpq
```

you will see the **`ntpq`** command prompt:

```
ntpq>
```

Use the command

```
peers
```

to display the NTP peers which your computer is using. One of them should be the Unison server which you have just configured. You should verify that it is being ‘reached’. (You may have to continue issuing the peers command for a minute or two before you will see the ‘reach’ count increment.)

If you are using authentication, you can verify that authentication is being used by issuing the command

```
associations
```

to display the characteristics of the client server associations. In the “auth” column of the display, you should see “OK” for the row corresponding to the Unison server. If you see “bad”, you should wait a few minutes to be sure that there is a problem since “bad” is the initial state of this setting. If the “bad” indication persists then you must check your configuration for errors. Typically this is due to a typing error in creating the */etc/ntp.keys* file on the client that causes a mismatch between the keys being used by the server and client. (If you transfer the file by **`ftp`** or **`scp`**, this shouldn’t be a problem.) It is also possible to have a typing error in the */etc/ntp.conf* file that causes the needed key to not be included in the “trustedkey” list.

ChapterFour

Setting Up NTP Clients on Windows NT 4.0/2000/XP

To configure your Windows NT 4.0/2000/XP computer to use your Unison, you must have successfully completed the Basic Installation procedures in Chapter 2. This manual is not a 'How-To' on installing and using NTP; basic approaches to NTP configuration for operation with the Unison will be described here. Installation must be performed by a user with administrative priviledges on the system. If you have never used NTP, then you should spend some time reading the on-line documents at: http://www.ntp.org.

Although all the information is available at the above site, the following are excellent tutorials on setting up NTP and are easier to understand:

http://www.sun.com/solutions/blueprints/0701/NTP.pdf
http://www.sun.com/solutions/blueprints/0801/NTPpt2.pdf
http://www.sun.com/solutions/blueprints/0901/NTPpt3.pdf

If you have a new server, many problems may be solved by the helpful people who participate in the Internet news group devoted to NTP at: comp.protocols.time.ntp.

Three methods of using the Unison with NTP clients on Window NT 4.0 platforms will be described:

Basic: This is the simplest, and will operate without MD5 authentication. **NTP beginners should always perform this setup firs**t.

MD5: This method is trickier only because MD5 keys must be set up and distributed accurately to the NTP clients in a secure way. The Unison is factory configured to authenticate its replies to NTP MD5 clients using its default set of keys.

Broadcast/Multicast: This method simplifies configuration of the clients on large networks since specific server addresses need not be configured in each client's *\winnt\system32\drivers\etc\ntp.conf* file. It can be configured either with or without MD5 authentication. However, it is highly recommended that authentication be configured when using broadcast /multicast mode due to the relative ease with which a fake NTP server can take over the clock setting of the broadcast/multicast clients on the network.

Basic NTP Client Setup

Basic setup is relatively simple, if:

- You have been able to successfully communicate with the Unison on your network.
- You have installed NTP on your client computer.

Configure NTP

You must edit the *ntp.conf* file which **ntpd.exe**, the NTP daemon, looks for by default in the the *\winnt\system32\drivers\etc* directory of the boot partition. If your NTP installation placed this file in a different place, you must find it and edit it. For example, XP uses *\windows\system32\drivers\etc*. Add this line to the *ntp.conf* file:

```
server 192.168.1.245
```

This line tells **ntpd.exe** to use the NTP server at address 192.168.1.245 in addition to any other servers which might also be configured in the *ntp.conf* file.

Re-start **ntpd.exe** to have it begin using the Unison server. By default, the NTP installation program installs **ntpd.exe** as a service called Network Time Protocol, and starts it. You must use the Services utility in Control Panel to stop the Network Time Protocol service and then re-start it.

Use the NTP utility **ntpq.exe** to check that **ntpd.exe** is able to communicate with the Unison. By default it is installed in the *\Program Files\Network Time Protocol* sub-directory of your Windows NT/2000/XP partition. From a console window, after issuing the command

```
ntpq
```

you will see the **ntpq.exe** command prompt:

```
ntpq>
```

Use the command

```
peers
```

to display the NTP peers which your computer is using. One of them should be the Unison server which you have just configured. You should verify that it is being ‘reached’. (You may have to continue issuing the peers command for a minute or two before you will see the ‘reach’ count increment.) If you have other peers configured, verify that the offset information for the Unison server peer and your other peers is in agreement to within a few milliseconds, assuming that the other peers are synchronized to that level of accuracy.

It may also be useful to start the NTP daemon in ‘debug’ mode (**ntpd -d**) to confirm successful configuration. The debug version of the NTP daemon is located in the *debug* sub-directory of your NTP directory. Refer to the NTP documentation for detailed usage of these debug utilities.

MD5 Authenticated NTP Client Setup

MD5 authenticated setup is relatively simple, if:

- You have been able to successfully communicate with the Unison on your network.

- Your Unison has been configured to perform authentication either by factory default, or by running the **`ntpconfig`** shell script. The example Unison authentication configuration shown in ***Chapter 2 - Configuring the Network Time Protocol*** will be assumed in the example configuration commands shown here.

- You have installed NTP on your client computer.

- You have successfully performed the ***Basic NTP Client Setup*** on your client computer.

Create the ntp.keys File

You must create a file named *ntp.keys* in the *\winnt\system32\drivers\etc* directory or, for XP, the *\windows\system32\drivers\etc* directory. It must be a copy of the one residing in the */etc* directory of your Unison. You can **`telnet`** into your Unison and start an **`ftp`** session with your client computer to send the Unison */etc/ntp.keys* file to your client computer, or use the secure copy utility **`scp`**, or use a text editor to create the equivalent file. Although you should first test your setup using the factory default */etc/ntp.keys* file in your Unison server, you should create your own keys after you understand the process and have your clients operating correctly with the default file.

IMPORTANT

Handling of the ***\windows\system32\drivers\etc\ntp.keys*** file is the weak link in the MD5 authentication scheme. It is very important that it is owned by “administrator” and not readable by anyone other than “administrator”.

After transferring the file, make sure that its security properties are set such that it is readable only by the “administrator”.

Configure NTP

You must edit the *ntp.conf* file which **`ntpd.exe`**, the NTP daemon, looks for by default in the the *\winnt\system32\drivers\etc* directory. If your NTP installation placed this file in a different place, you must find it and edit it. For example, XP uses *\windows\system32\drivers\etc*. Add these lines to the end of the *ntp.conf* file:

```
keys \winnt\system32\drivers\etc\ntp.keys
trustedkey 1 2
```

Modify the line added previously in ***Basic NTP Client Setup*** so that authentication will be used with the Unison server using one of the trusted keys, in this case key # 1:

```
server 192.168.1.245 key 1
```

Re-start **`ntpd.exe`** to have it begin using the Unison server with MD5 authentication. By default, the NTP installation program installs **`ntpd.exe`** as a service called Network Time Protocol, and starts it. You must use the Services utility in Control Panel to stop the Network Time Protocol service and then re-start it.

Use the NTP utility **`ntpq.exe`** to check that **`ntpd.exe`** is able to communicate with the Unison. By default it is installed in the *\Program Files\Network Time Protocol* sub-directory of your Windows NT/2000/XP partition. From a console window, after issuing the command

```
ntpq
```

you will see the **`ntpq.exe`** command prompt:

```
ntpq>
```

Use the command

```
peers
```

to display the NTP peers which your computer is using. One of them should be the Unison server which you have just configured. You should verify that it is being 'reached'. (You may have to continue issuing the peers command for a minute or two before you will see the 'reach' count increment.)

You can verify that authentication is being used by issuing the command

```
associations
```

to display the characteristics of the client server associations. In the "auth" column of the display, you should see "OK" for the row corresponding to the Unison server. If you see "bad", you should wait a few minutes to be sure that there is a problem since "bad" is the initial state of this setting. If the "bad" indication persists then you must check your configuration for errors. Typically this is due to a typing error in creating the *\winnt\system32\drivers\etc\ntp.keys* file on the client that causes a mismatch between the keys being used by the server and client. (If you transfer the file by **`ftp`** or **`scp`**, this shouldn't be a problem.) It is also possible to have a typing error in the *\winnt\system32\drivers\etc\ntp.conf* file that causes the needed key to not be included in the "trustedkey" list.

Broadcast/Multicast NTP Client Setup

Broadcast/multicast client setup is relatively simple, if:

- You have been able to successfully communicate with the Unison on your network.

- Your Unison has been configured to perform broadcasts or multicasts by running the **`ntpconfig`** shell script. (This is not the factory default configuration, so be sure to run **`ntpconfig`**.) If you are going to use MD5 authentication, your Unison must have been configured to operate with authentication in the broadcast/multicast mode, and you must know which of the trusted keys it is using for broadcast/multicast operation. The example Unison configuration shown in ***Chapter 2 - Configuring the Network Time Protocol*** will be assumed in the example configuration commands shown here.

- You have installed NTP on your client computer.

- You have successfully performed the ***MD5 Authenticated NTP Client Setup*** on your client computer, if you plan to use MD5 authentication.

Configure NTP

You must edit the *ntp.conf* file which **`ntpd.exe`**, the NTP daemon, looks for by default in the the *\winnt\system32\drivers\etc* directory or, for XP, the *\windows\system32\drivers\etc* directory. Assuming that your Unison server has been configured to use key 2 for broadcast authentication as shown in the example in chapter 2, make sure that key 2 is included in the **`trustedkey`** line, and add this line to the end of the *ntp.conf* file:

```
broadcastclient
```

If you are not using MD5 authentication, you would add these lines:

```
disable auth
broadcastclient
```

If you are using multicast instead of broadcast mode, you would replace the **`broadcastclient`** keyword with the **`multicastclient`** keyword. You may remove the line added previously in ***Basic NTP Client Setup***:

```
server 192.168.1.245
```

or the authenticated version added in ***MD5 Authenticated NTP Client Setup***:

```
server 192.168.1.245 key 1
```

Re-start **`ntpd.exe`** to have it begin using the Unison as a broadcast or multicast server. By default, the NTP installation program installs **`ntpd.exe`** as a service called Network Time Protocol, and starts it. You must use the Services utility in Control Panel to stop the Network Time Protocol service and then re-start it.

Use the NTP utility **`ntpq.exe`** to check that **`ntpd.exe`** is able to communicate with the Unison. By default it is installed in the *\Program Files\Network Time Protocol* sub-directory of your Windows NT/2000/XP partition. After issuing the command

```
ntpq
```

you will see the **`ntpq.exe`** command prompt:

```
ntpq>
```

Use the command

```
peers
```

to display the NTP peers which your computer is using. One of them should be the Unison server which you have just configured. You should verify that it is being ‘reached’. (You may have to continue issuing the peers command for a minute or two before you will see the ‘reach’ count increment.)

If you are using authentication, you can verify that authentication is being used by issuing the command

```
associations
```

to display the characteristics of the client server associations. In the “auth” column of the display, you should see “OK” for the row corresponding to the Unison server. If you see “bad”, you should wait a few minutes to be sure that there is a problem since “bad” is the initial state of this setting. If the “bad” indication persists then you must check your configuration for errors. Typically this is due to a typing error in creating the *\windows\system32\drivers\etc\ntp.keys* file on the client that causes a mismatch between the keys being used by the server and client. (If you transfer the file by **`ftp`** or **`scp`**, this shouldn’t be a problem.) It is also possible to have a typing error in the *\windows\system32\drivers\etc\ntp.conf* file that causes the needed key to not be included in the “trustedkey” list.

Chapter*Five*

Control and Status Commands

This chapter describes the Unison control and status commands. The Unison supports several application-specific commands for performing initialization/setup and for monitoring the performance and status of the NTP and CDMA subsystems. You do not need knowledge of Linux commands in order to operate the Unison. However, the Unison 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 Unison-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 Unison. However, for those interested, the command shell used by the Unison 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 Unison 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 Unison 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.
cdmachannelset	Prints the current CDMA channelset being used. It can be one of North American Cellular, South Korean Cellular, North American PCS, Indian Cellular or Japanese Cellular.
cdmaleapconfig	Guides the user in configuring the way in which UTC leap seconds are handled: either automatically via CDMA basestation transmissions or by user-entered current and future leap second parameters.
cdmaleapmode	Prints the current CDMA leap second mode of operation, either automatic or user-entered. If user-entered, prints the current and future leap second values.
cdmastat	Prints the CDMA subsystem status information to the console.
cdmaversion	Prints the CDMA firmware and FPGA version information to the console.
cntphwaddr	Prints the ethernet hardware address, if the ethernet has been configured.
cntposctype	Prints the installed oscillator type which is TCXO or MS-OCXO.
cntppasswd	Allows the *root* user to change the password for the two configured users on the Unison: *cntpuser* and *root*. This script calls the standard Linux **`passwd`** binary and then saves the resulting */etc/shadow* file to the non-volatile FLASH disk.
cntprootfs	Prints the current root file system image, either UnisonCDMA_0 (factory default) or UnisonCDMA_1 (field upgrade) which is running in the Unison to the console.
cntpstat	Parses the output of **`ntpq –c peers`** to obtain the system peer status of the NTP CDMA reference clock. It also retrieves the current reference clock polling status data and prints it to the console.
cntptimemode	Prints the time mode settings in effect for any optional timecode or Serial Time output.
cntptimemodeconfig	Interactive shell script that guides the user in configuring the time mode settings for any optional timecode output. Allows setting to the local, GPS or UTC timescale. If local-manual is selected, then the allows configuration of the local offset and Daylight Savings Time (DST) start/stop date parameters.
cntpversion	Prints the Unison application software version information to the console.
cpuopts	Returns the current settings for any installed, user-selectable, CPU Options. These are: 1PPS, AM Code or Prog TTL.

cpuoptsconfig	An interactive script that allows the user to modify the settings for the CPU Options listed above.
cpusertime	Prints the current settings for the optional Serial Time output.
cpusertimeconfig	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/NTP upgrade process.
help	Prints help for Unison 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 Unison.
netconfig	Interactive shell script that allows the user to configure the IP network subsystem of the Unison.
ntpconfig	Interactive shell script that guides the user in configuring the Unison NTP subsystem. Allows configuration of MD5 authentication and broadcast/multicast mode. All parameters are retained in non-volatile FLASH disk storage.
ptpconfig	Interactive shell script that guides the user in configuring parameters for the PTP/IEEE-1588 protocol. This command is only available if the PTP/IEEE-1588 option has been installed.
setcdmachannelset	Command that allows the user to select the channelset for the CDMA sub-system to receive. This command is not functional in units comfigured for Japanese Cellular operation.
setsigfltmask	Command to mask or enable the Signal Loss Fault.
sigfltmask	Prints the current setting for the Signal Loss Fault mask.
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.
upgradecdma	Shell script that facilitates the CDMA 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 Unison. 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 Unison after running this script for the changes to take effect.

Set: **`accessconfig`**
Unison response: Interactive shell script is started.

cdmachannelset

This command displays the CDMA channelset currently being used by the CDMA subsystem. It can be one of: North American Cellular, South Korean Cellular, North American PCS, Indian Cellular, or for certain units, Japanese Cellular.

Query: **`cdmachannelset`**
Unison response: **`Channelset is North American PCS`**

cdmaleapconfig

Leap seconds affect NTP, UTC and Local Time (not GPS Time). Leap second insertions occur about once every two years. This command starts an interactive shell script that will guide the root user in configuring the way that UTC leap seconds are handled.

There are two different modes for handling leap second insertions: automatic and user-entered. The Unison is shipped from the factory in user-entered leap second mode with the current and future leap second values set appropriately. You will need to change these values the next time a leap second is pending. The interactive script is very detailed in explaining how these values are obtained and used. There is also more information in ***Appendix D - Leap Seconds***.

Query: **`cdmaleapconfig`**
Unison response: Interactive shell script is started.

cdmaleapmode

This command displays the CDMA leap mode of operation currently configured. There are two modes: automatic and user-entered. If the mode is user-entered, then the values of the configured current and future leap seconds are also displayed.

Query: **`cdmaleapmode`**
Unison response:

```
CDMA Leap Second Mode is USER:  Current LS = 13, Future LS = 13
```

cdmastat

This command allows the user to query the status of the CDMA timing subsystem. During normal operation, the NTP daemon polls the CDMA 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 data contained therein and prints this fixed-length string having these fields:

```
LKSTAT TFOM = ? YEAR DOY HH:MM:SS.ssssssss LS S C PNO AGC VCDAC SN.R F.ERR FLTS
```

Where:

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

TFOM = ? A detailed explanation of TFOM is in ***Appendix E - Time Figure-of-Merit***. Briefly, TFOM indicates clock accuracy where:

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 CDMA.

YEAR is the year of the UTC timestamp of the most recent NTP polling request received by the CDMA subsystem from the NTP reference clock driver.

DOY is the day-of-year of the UTC timestamp of most recent NTP polling request received by the CDMA subsytem from the NTP reference clock driver.

HH:MM:SS.sssssssss is the hour, minute, second.subsecond UTC timestamp of the most recent NTP polling request received by the CDMA subsystem from the NTP daemon reference clock driver.

LS is the current 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 (Signal Detected), 2 (Code Locking), 4 (Carrier Locking), 8 (Locked).

C is the CDMA frequency channel being used, one of 0 thru 61, depending upon the channelset being used:
For North American Cellular these are:
0 (Primary A), 1 (Primary B), 2 (Secondary A), 3 (Secondary B)
For South Korean Cellular these are:
4 (Primary A), 5 (Primary B), 6 (Secondary A), 7 (Secondary B)
For North American PCS these are:
8 (00 A), 9 (01 A), 10 (02 A), 11 (03 A), 12 (04 A), 13 (05 A), 14 (06 A), 15 (07 A),
16 (08 A), 17 (09 A), 18 (10 A),
19 (00 D), 20 (01 D), 21 (02 D) ,
22 (00 B), 23 (01 B), 24 (02 B), 25 (03 B), 26 (04 B), 27 (05 B), 28 (06 B), 29 (07 B),
30 (08 B), 31 (09 B), 32 (10 B)
33 (00 E), 34 (01 E), 35 (02 E)
36 (00 F), 37 (01 F), 38 (02 F)
39 (00 C), 40 (01 C), 41 (02 C), 42 (03 C), 43 (04 C), 44 (05 C),
45 (06 C), 46 (07 C), 47 (08 C), 48 (09 C), 49 (10 C).
For Japanese Cellular these are:
50 (Primary A), 51 (Secondary A)
For Indian Cellular these are:
52 (Sri Lanka 43), 53 (Sri Lanka 146),
54 (India 185), 55 (India 226), 56 (India 267), 57 (India 308),
58 (India 369), 59 (India 410), 60 (India 451), 61 (India 492).

PNO is the base station pseudonoise offset, 0 to 511 in units of 64 pseudonoise code chips.

AGC is the automatic gain control DAC byte, 0 to 255 with larger numbers implying higher RF gain. Typical range is 150 to 220.

VCDAC is the upper 16 bits of the TCXO voltage control DAC word, 0 to 65535 with larger numbers implying higher TCXO frequency. Typical range is 20000 to 38000.

SN.R is the carrier signal-to-noise ratio, 0.00 to 99.9, measured in the CDMA sync channel symbol rate bandwidth. Typical range is 2.5 to 11.0.

F.ERR is the CDMA sync channel frame error rate, 0.000 to 1.000, with a higher number implying more Cyclical Redundancy Check (CRC) failures when processing the sync channel message frames. Higher numbers will correlate with lower signal-to-noise ratios.

FLTS is the fault status, which displays the current summary status of the CDMA 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	Not Used	No Polling Events	Time Input Fault	Local Osc 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 CDMA 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.

No Signal Time-Out: This bit indicates that the unit has not been able to acquire a CDMA 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.

Local Oscillator Fault: This bit indicates that the receiver Local Oscillator Phase Locked Loop (PLL) synthesizer is either unlocked or has failed. This condition should not normally occur unless the unit is subjected to out-of-specification environmental conditions. Otherwise, this would be a fatal fault and the unit should be returned to the factory for repair (1-877-749-3878).

Time Input Fault: This bit indicates that the microprocessor received an erroneous time input from the CDMA engine. If the condition persists please report it to the factory.

No Polling Events: This bit indicates that the CDMA timing subsystem is not receiving polling request from the NTP subsystem. 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.

The example response indicates that there has been a period without tracking a CDMA signal that exceeded the time-out period, that there was a FLASH Write Fault and that there is a Local Oscillator PLL fault.

Query: **`cdmastat`**
Unison response:

```
LOCKED TFOM = 6 2001 092 04:48:56.347916732 13 8 1 132 28605 8.6 0.000 001A
```

cdmaversion

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

Query: **`cdmaversion`**
Unison response: **`F/W 1.00 FPGA 0202`**

cntphwaddr

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

Query: **`cntphwaddr`**
Unison response: **`00:0E:FE:00:00:33`**

cntposctype

This command displays the installed oscillator type. It is TCXO or MS-OCXO. The standard oscillator is the TCXO.

Query: **`cntposctype`**
Unison response: **`Installed Oscillator is TCXO`**

cntppasswd

This command allows the root user to change the passwords of the two configured users on the system: *root* and *cntpuser*. Arguments passed to **`cntppasswd`** 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.

To change root password:
Set: **`cntppasswd`**
Unison response: The passwd interactive utility starts.

To change cntpuser password:

Set:	**`cntppasswd cntpuser`**
Unison response:	The passwd interactive utility starts.

cntprootfs

This command displays the currently booted root file system image. It can be either UnisonCDMA_0 (factory image) or UnisonCDMA_1 (field upgrade image). Refer to ***Appendix B - Upgrading the Firmware*** for detailed instructions on performing the upgrade procedure.

Query:	**`cntprootfs`**
Unison response:	**`BOOT_IMAGE=UnisonCDMA_1`**

cntpstat

This command allows the user to query the status of the NTP subsystem. It retrieves information from the NTP distribution **`ntpq`** binary using the *peers* command to determine the current synchronization status of the NTP subsystem. It then retrieves the last line in the logfile */var/log/praecis0.monitor* controlled by the NTP daemon reference clock driver that communicates with the CDMA timing subsystem. This logfile is updated every 16 seconds under normal operation. It parses and formats the data contained therein and prints this fixed-length (generally, grossly unsynchronized states could cause the floating offset field to overflow momentarily) string having these fields:

```
LKSTAT TO CDMA, Offset = +S.ssssss, TFOM = ? @ YEAR DOY HH:MM:SS.sssssssss LS
```

Where:

LKSTAT is the system peer status of the NTP daemon relative to the CDMA subsystem engine, either LOCKED or NOTLKD. Not locked can imply several things: the system has just started, there is a fault in the CDMA subsystem which has caused NTP to either be unable to obtain timing information from the CDMA subsystem or to reject the timing information that it is obtaining from it.

+S.ssssss is the offset in seconds between the NTP system clock and the CDMA subsystem clock. Positive implies that the system clock is ahead of the CDMA subsystem clock.

TFOM = ? A detailed explanation of TFOM is in ***Appendix E - Time Figure-of-Merit***. Briefly, TFOM indicates clock accuracy where:

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 CDMA.

YEAR is the year of the UTC timestamp of most recent NTP polling request received by the CDMA engine from the NTP reference clock driver.

DOY is the day-of-year of the UTC timestamp of most recent NTP polling request received by the CDMA engine from the NTP reference clock driver.

HH:MM:SS.sssssssss is the hour, minute, second.subsecond UTC timestamp of the most recent NTP polling request received by the CDMA engine from the NTP daemon reference clock driver.

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

Query: `cntpstat`

Unison response:

```
LOCKED TO CDMA, Offset = +0.000024, TFOM = 6 @ 2001 092 06:03:10.904312858 13
```

cntptimemode

This command displays the current time mode setting for any optional timecode outputs. Possible time modes are: UTC, GPS, Local-Auto and Local-Manual. The local time offset from UTC is valid in either of the two local modes, but the Daylight Savings Time (DST) start/stop parameters are only valid in the local-manual mode. A positive local time offset implies a longitude east of the Greenwich meridian and that local time is ahead of UTC.

In the local-automatic mode, the local offset from UTC is determined from the CDMA base station transmissions. For more precise and deterministic behavior at the DST changeover times, you should configure your unit for local-manual operation and set up the local offset and the DST start and stop times using `cntptimemodeconfig`.

Query: `cntptimemode`

Unison response:

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

cntptimemodeconfig

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

By default, the unit is configured to operate in UTC time mode. If you need to modify this operation, you must run this script as *root*. Settings made using this command are non-volatile.

Set: `cntptimemodeconfig`

Unison response: Interactive shell script is started.

cntpversion

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

Query: `cntpversion`

Unison response:

```
Unison CDMA 6010-0042-000 v 1.00 Fri Aug 20 22:38:21 UTC 2004
```

cpuopts

This command displays the current settings for the installed CPU Options.

Query: `cpuopts`

Unison response:

```
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 Options. The user-selectable options are: 1PPS, AM Code, and Prog TTL.

Set:	**`cpuoptsconfig`**
Unison response:	Interactive shell script is started.

cpusertime

This command displays the current settings for the optional Serial Time Output. Settings for the three NMEA Sentences are always shown but are only applicable if you have selected NMEA as the Output Format Setting. More information about the various formats is in ***Appendix G - Serial Time Output***.

Query: **`cpusertime`**

Unison response:

```
Current Serial Time Output Baud Rate Setting = 9600
Current Serial Time Output Format Setting = Sysplex
Current Serial Time Output Parity Setting = Odd
Current NMEA Sentence 1 Setting - ZDA
Current NMEA Sentence 2 Setting - NONE
Current NMEA Sentence 3 Setting - NONE
```

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, NENA0, NENA1, NENA8 and NMEA), the baud rate (4800, 9600, 19200, 57600) and the parity (ODD, EVEN, or NONE). The three NMEA sentences can also be changed but are applicable only if the Output Format is NMEA. More information about the various formats is in ***Appendix G - Serial Time Output***.

Set:	**`cpusertimeconfig`**
Unison response:	Interactive shell script is started.

eraserootfs_1

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

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

help

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

Query:	**`help`**
Unison response:	Tempux LX commands are displayed.
Query:	**`help cdmastat`**
Unison response:	Information specific to the **`cdmastat`** 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 Unison. Three 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 Unison after running this script for the changes to take effect.

Set:	**`inetdconfig`**
Unison 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 Unison. 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 Unison after running this script for the changes to take effect.

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

ntpconfig

This command starts an interactive shell script that will allow the user to configure the NTP subsystem of the Unison. By default, the unit is configured to authenticate its replies to clients using its default MD5 keys in the */etc/ntp.keys* file. If you need to create your own MD5 keys (recommended) or set up broadcast/multicast operation, you must run this script as root. Refer to ***Chapter 2 - Configuring the Network Time Protocol*** for details on the use of this command.

The two files that are modified are */etc/ntp.keys* and */etc/ntp.conf*. Both of these are non-volatilely stored in the FLASH disk */boot/etc* directory. You must re-boot the Unison after running this script for the changes to take effect.

Set:	**`ntpconfig`**
Unison response:	Interactive shell script is started.

ptpconfig (Optional)

This command is only available if the PTP/IEEE-1588 option has been installed. Refer to ***Appendix H - Precision Time Protocol/IEEE-1588*** for more information.

setcdmachannelset

This command sets the CDMA channelset to be used by the CDMA subsystem. By factory default, the channelset is North American Cellular, unless the unit is configured for Japanese Cellular operation. In that case the hardware configuration limits operation to only the Japanese Cellular band, and this command will have no affect. The command requires one argument, which may be one of these strings: NAC (North American Cellular), SKC (South Korean Cellular), NAP (North American PCS), or IND (Indian Cellular).

Set:	**`setcdmachannelset NAP`**
Unison response:	**`Channelset is North American PCS`**

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`**
Unison response:	**`Signal Loss Fault Mask set to MASKED`**

sigfltmask

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

Query:	**`sigfltmask`**
Unison response:	**`Signal Loss Fault is ENABLED`**

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 Unison 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`**
Unison response:	**`UPGRADE is the default root file system.`**
Query:	**`updaterootflag 2`**
Unison response:	**`UPGRADE is the default root file system.`**

upgradecdma

This script allows the user to upgrade the CDMA subsystem firmware. It requires one argument: the path to the binary file to be uploaded to the CDMA engine. It issues the commands over the serial port to the CDMA subsystem that are needed to start the X-modem file transfer, and then displays the responses from the CDMA subsystem to the console. When the X-modem 'C' character appears, indicating that the CDMA 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 CDMA 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 **`upgradecdma`** script. All other lines are from the CDMA 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 CDMA subsystem as it comes up. The firmware version should match that of the binary file that was uploaded. See ***Performing the CDMA Upgrade*** in ***Appendix B - Upgrading the Firmware*** for more information.

Set: **`upgradecdma /tmp/6010-0020-000.bin`**
Unison 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/upgradecdma: line 26: 27618 Terminated            cat </dev/arm_user

---You should see the CDMA 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 Unison. Changing the Linux kernel firmware will enable IPv6 operation and should only be done if you have a requirement for IPv6. See ***Chapter 6 - IPv6 Information*** and ***Performing the Linux Kernel Upgrade*** in ***Appendix B - Upgrading the Firmware*** for more information.

Set **`upgradekernel /tmp/newkernelimage`**
Unison 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 Unison is wired as shown below. In order to connect the Unison 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 Unison to your computer.

Unison 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 Six

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 Unison check the firmware version using console port command **`cat /proc/version`**.

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

```
6010-0041-000 ver 2.4.31-IPv4
```

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. This 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 or 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 *A*

Security

Your Unison 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 Unison. Others are provided by the additional protocol servers selected for inclusion in your Unison, 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. In addition, the Network Time Protocol daemon,* **`ntpd`** *supports client-server authentication security measures to deter spoofing of NTP clients by rogue NTP servers. 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 Unison 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 Unison.

- 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 Unison 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 Unison 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 Unison 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 Unison. This means that users on remote hosts may log in to the Unison via an **ssh** client, but users logged in on the Unison 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 Unison via **scp** over **ssh** but users logged in on the Unison 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 Unison 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 Unison. 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 Unison. 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 Unison. At boot time, the Unison 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 Unison. 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.

Network Time Protocol

The NTP implementation in the Unison is built from the standard distribution from the http://www.ntp.org site. By factory default, remote control of the NTP daemon **ntpd** is disabled. Query-only operation is supported from the two NTP companion utilities **ntpq** and **ntpdc**.

Control via these two utilities is disabled in the */etc/ntp.conf* file in two ways. First, MD5 authentication keys are not defined for control operation via a *requestkey* or *controlkey* declaration. Second, this default address restriction line is present in the file:

restrict default nomodify

This line eliminates control access from ALL hosts. Query access is not affected by this restriction. Knowledgable NTP users who would like to customize the security aspects of the configuration of the NTP daemon in the Unison should edit the */etc/ntp.conf* file directly and then copy it to the */boot/etc* directory. Be sure to retain the ownership and permissions of the original file by using **cp -p** when performing the copy.

CAUTION

If you are planning to make changes to the ***/etc/ntp.conf*** file, you must not restrict query access from the local host to the NTP daemon. Various system monitoring processes running on the system require this access.

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 Unison.

The firmware consists of two FLASH binary image files. One of these is the firmware for the Unison itself. This firmware executes on the IBM-compatible CPU and contains the embedded Linux operating system and NTP specific application software. The other file is the firmware for the CDMA 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 Unison. This means that you must place the previously downloaded file(s) in a place on your network which is accessible to the Unison.

Performing the Linux/NTP 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 file when downloading from the website. Please make sure that the downloaded file size matches what the website says it should be. Upgrading the partition with a too-large file size will cause problems.

To perform the upgrade, log in as the *root* user to the Unison 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 Unison using FTP. The root file system image will be named with the software part number and version like: *6010-004x2-000_3.00.gz*. When following the instructions below, substitute the name of the actual root file system image that you are installing for *6010-004x-000_3.00.gz*. Issue these commands from the console of your Unison:

```
ftp remote_host                        {perform ftp login on remote host}
bin                                    {set transfer mode to binary}
get 6010-004x-000_3.00.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 Unison. A command like this should be used:

```
scp -p 6010-004x-000_3.00.gz root@cntp.your.domain:/dev/rootfs_1
```

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

```
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 Unison 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

```
cntpversion
```

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:

```
cntprootfs
```

Which should cause this to be printed to the console:

```
BOOT_IMAGE=UnisonCDMA_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 Unison.

Recovering from a Failed Upgrade

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

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 Unison 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 Unison using FTP. The IPv6 kernel image will be named with the software part number like: *6010-0041-100.bzimage*. When following the instructions below, substitute the name of the actual kernel image that you are installing for *6010-0041-100.bzimage.* Issue these commands from the console of your Unison:

```
ftp remote_host                         {perform ftp login on remote host}
bin                                     {set transfer mode to binary}
get 6010-0041-100.bzimage /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 Unison. A command like this should be used:

```
scp –p 6010-0041-100.bzimage 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/6010-0042-100.bzimage
```

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 CDMA Upgrade

To perform this upgrade, log in as the *root* user to the Unison 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* . The CDMA subsystem image will be named with the software part number and version like: *6010-0020-000_3.01.bin*. When following the instructions below, substitute the name of the actual CDMA subsystem image that you are installing for *6010-0020-000_3.01.bin*:

```
ftp remote_host                {perform ftp login on remote host}
bin                            {set transfer mode to binary}
get 6010-0020-000_3.01.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 CDMA subsystem image to the */tmp* directory using **scp** from the remote computer. A command like this could be used:

```
scp -p 6010-0020-000_3.01.bin root@cntp.your.domain:/tmp
```

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

```
upgradecdma /tmp/6010-0020-000_3.01.bin
```

This command is a script that performs the file transfer to the CDMA engine. It first tells the CDMA 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 CDMA engine to the console. You should next see this message from the CDMA 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/upgradecdma: line 26: 27618 Terminated       cat </dev/arm_user

---You should see the CDMA 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 CDMA 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 CDMA 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 **upgradecdma** script terminates its execution, and you will again have the standard Unison console prompt.

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

```
cdmaversion
```

The upgraded version information should be displayed.

Problems with the CDMA 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 CDMA 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 CDMA 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:

```
upgradecdma /tmp/6010-0020-000_3.01.bin
```

Recover Command

Sometimes a user will attempt to download the wrong file to the CDMA Subsystem. When this happens the recovery method above will not work. After issuing the **`cat`** command above you will not see a series of “C” characters, but instead you will see the bootloader message being output every few seconds. In this case you need to use a different recovery procedure.

First make sure the above **`cat`** command is killed by typing CTRL-C. Then enter a new **`cat`** command as:

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

You should again be seeing the bootloader message every few seconds:

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

Please type the following command but do not press enter:

```
echo -e “recover\r” > /dev/arm_user
```

Now wait until you see another bootloader message come out and then press enter. You will then see the “C” come out every 3 seconds. You then kill the previous **`cat`** command by entering:

```
kill $!
```

You should see a command prompt. Now issue this command to re-transfer the correct binary file:

```
upgradecdma /tmp/6010-0020-000_3.01.bin
```

Appendix C

Simple Network Management Protocol (SNMP)

Your Unison 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:

TEMPUSLXUNISON-MIB

Which is located on your Unison in this ASCII file:

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

In addition to a complete set of NTP and CDMA status objects, the MIB defines four SMIv2 notification objects:

- NTP Leap Indicator Bits status change
- NTP Stratum change
- CDMA Fault Status change
- CDMA 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 TEMPUSLXUNISON-MIB file on your SNMP management system and access the variables defined therein. The factory default community names are “TempusLXUnison” 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   TempusLXUnison
```

Configuring SNMPv1 Trap Generation

To have your Unison 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 Unison. 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 Unison will only send a trap to the last declared **`trapsink`** in the file.

Configuring SNMPv2c Notifications and Informs

To have your Unison 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 Unison. 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 Unison 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 Unison 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 Unison, 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 ntpuser 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 *ntpuser* 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).endRun-TechnologiesMIB(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 ntpuser SHA Tempus_0
```

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 *ntpuser* to be created who may be authenticated using the Secure Hash Algorithm (SHA) with password *Tempus_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/snmp.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*

Leap Seconds

The Network Time Protocol (NTP) uses UTC (Coordinated Universal Time) for its time mode. UTC is affected by leap seconds. This is an additional second that is inserted into UTC in order to keep it in agreement with the Earth's rotation. Leap seconds may only be inserted at UTC midnight on June 30th or on December 31st. Leap second insertions (or transitions) occur about every 2 years.

Your Unison can automatically get the leap second information from the CDMA transmissions. However, some of the CDMA providers have not implemented this to the level of precision needed for a perfectly smooth transition at UTC midnight on the day of a leap second insertion. To ensure that your Unison will precisely handle any UTC leap second transitions, your unit has been configured at the factory for the user-entered leap mode.

If you prefer to have your Unison automatically get its leap second information from the CDMA transmissions then just change the leap mode to automatic. You can do this by using console command **`cdmaleapconfig`** (see ***Chapter 5 - Control and Status Commands***). To view the leap second settings use command **`cdmaleapmode`**.

In the user-entered leap mode, the current and future leap second values must be manually set. You can do this by using console command **`cdmaleapconfig`**. When the unit is configured at the factory, the current and future leap second values are set appropriately for the next possible leap second insertion date (June 30th or December 31st). If there is no leap second insertion scheduled, then the same value is set for both the current and future leap seconds. If there is a leap second insertion scheduled, then a future value is set that is one more than the current value.

For example, as of this writing (July 2005) there are 13 leap seconds. The next possible leap second insertion date is December 31st and there WILL be a leap second insertion on that date. After December 31st there will be 14 leap seconds. So, all units shipped from the factory between July 2005 and the end of December 2005 will have a current leap second value of 13 and a future leap second value of 14. Your Unison will remember the settings and make a perfect leap second transition at midnight on December 31st. If you happen to have your unit powered off on December 31st, then when power is reapplied, it will know that the leap second transition has passed and behave appropriately. After December 31st, your unit will show that both the current and future leap seconds are 14. This will continue as long as there is no new leap second insertion scheduled, probably for many years. When a new leap second insertion is scheduled you will need to alter the current and future values.

The EndRun Technologies' website has a page devoted to notifying users of the next leap second occurrence. It also posts the appropriate current and future leap seconds setting for your Unison. The appropriate link is:

http://www.endruntechnologies.com/leap.htm

Background Information

Another way to get the leap second information is to go to the International Earth Rotation Service (IERS) website. If a leap second is pending it will be posted by the IERS approximately six months in advance of insertion. This information is available in the latest Bulletin C at the (IERS) website:

http://www.iers.org

Leap seconds are inserted from time-to-time in order to keep UTC, which is derived from atomic time (TAI), in agreement with the Earth's rotation rate. Relative to TAI, the Earth's rotation rate is slowing down. This means that UTC must be retarded periodically in order to maintain agreement between UTC and the apparent daylength. If this were not done, eventually UTC would drift out-of-sync with Earth's day and many astronomical and navigational problems would ensue.

The International Earth Rotation Service (IERS) is the organization responsible for measuring the relationship between UTC and the rotation rate of the Earth. When the difference between UTC and apparent Earth time has exceeded a certain threshold, the IERS coordinates with the Bureau International of the Hour (BIH) to schedule the insertion of a leap second into the UTC time scale.

The IERS publishes Bulletin C about 6 months in advance of each possible leap second insertion point. Leap seconds may only be inserted at UTC midnight of June 30 or December 31. Bulletin C confirms either that a leap second will or will not be inserted at the next possible insertion point. Since the introduction of leap seconds in 1961, they have been added approximately once every 18 months.

The leap seconds which are needed for your Unison are actually the difference between GPS-UTC. The GPS time scale began on January 6, 1980. At that time, the UTC timescale had already undergone 19 leapsecond insertion events. If you are obtaining your leap second information from the IERS website, you will need to subtract 19 from the TAI-UTC leap second values published there to obtain GPS-UTC, the number needed to set the current and future leap seconds for the Unison. At the time of this writing in July of 2005, TAI-UTC was 32 seconds and GPS-UTC was 13 seconds.

Appendix *E*

Time Figure-of-Merit (TFOM)

This appendix describes the Time Figure of Merit (TFOM) number. The Unison displays this number in the time-of-day fields printed by the Unison **`cdmastat`** *and* **`cntpstat`** *commands (see Chapter 5). The TFOM number indicates the level of accuracy that should be included in the interpretation of the time-of-day and ranges from 6 to 9:*

6	time error is < 100 microseconds
7	time error is < 1 milliseconds
8	time error is < 10 milliseconds
9	time error is > 10 milliseconds, unsynchronized state if never locked to CDMA

In all cases, the Unison reports this value as accurately as possible, even during periods of CDMA signal outage where the Unison is unable to directly measure the relationship of its timing outputs to UTC. During these CDMA outage periods, assuming that the Unison had been synchronized prior to the outage, the Unison extrapolates the expected drift of the Unison 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 CDMA signal reception will not induce an immediate alarm condition. 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 Unison is unable to achieve re-synchronization within one hour after reaching this state, the red LED will illuminate. The fault status field returned in either of the **`cdmastat`** or **`cntpstat`** commands will have the appropriate bit set to indicate a loss-of-signal timeout condition.

If the CDMA subsystem reaches the unsynchronized TFOM state, the NTP daemon will cease to use the timing information returned by the CDMA subsystem in its polling event timestamps. At this point, the NTP daemon will report in its replies to network NTP clients that it is running at stratum 16 and the leap indicator bits will be set to the fault state. NTP clients will recognize that and cease to use the unsynchronized server.

Appendix*F*

Third-Party Software

The Unison is running several different software products created and/or maintained by open source projects. Open source software comes with its own license. These are 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

NTP Software License

Information about the NTP Project, led by Dr. David Mills, can be found at www.ntp.org. The distribution and usage of the NTP software is allowed, as long as the following copyright notice is included in our documentation:

The following copyright notice applies to all files collectively called the Network Time Protocol Version 4 Distribution. Unless specifically declared otherwise in an individual file, this notice applies as if the text was explicitly included in the file.

Appendix G

Serial Time Output

This option is provided on a second RS-232 serial port. It is a serial time string 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. This output starts automatically at power-up.

To configure this output refer to ***Chapter 5 - Control and Status Commands*** for details on the **`cpusertime`** and **`cpusertimeconfig`**.

There are several different formats for this string. The format, baud rate and parity can all be changed via the front-panel keypad or via the 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, NENA and NMEA.

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 the string is UTC and it 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 Unison. (See **gntptimemodeconfig** in ***Chapter 5 - 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 time contained in this string depends on the time mode of the Unison. For example, if you want the time in this string to be UTC, then set the time mode of the Unison to UTC. (See **gntptimemodeconfig** in ***Chapter 5 - Control and Status Commands***.) 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 E - 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.

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 CC FF<CR><LF>
```

T	is the Time Figure of Merit (TFOM) character described in ***Appendix E - 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.
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 United States National Emergency Number Association. This organization has adopted several ASCII time code formats for use in PSAPs (Public Safety Answering Points) and they are specified in the NENA PSAP Master Clock Standard, Issue 4. These ASCII time code formats are NENA Format 0 (NENA0), NENA Format 1 (NENA1), and NENA Format 8 (NENA8).

NENA0

<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. This is the "on-time" character.
^	is the space character (0x20).
DDD	is the day-of-year (001-366)
:	is the colon character (0x3A)
HH	is the hour-of-the-day (00-23)
MM	is the minute-of-the-hour (00-59)
SS	is the second-of-the-minute (00-60)
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).

NENA1

<CR><LF>Q^WWW^DDMMMYY^HH:MM:SS<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. This is the "on-time" character.
^	is the space character (0x20).
WWW	is the day-of-week (MON, TUE, WED, THU, FRI, SAT
DD	is the day-of-month (1-31)
MMM	is the month (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC)
YY	is the two-digit year
:	is the colon character (0x3A)
HH	is the hour-of-the-day (00-23)
MM	is the minute-of-the-hour (00-59)
SS	is the second-of-the-minute (00-60)
<CR>	is the ASCII carriage return character (0x0D). The first <CR> is the on-time character.
<LF>	is the ASCII line feed character (0x0A)

NENA8

<CR><LF>Q^^YYYY^DDD^HH:MM:SS^D+ZZ<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. This is the "on-time" character.
^	is the space character (0x20).
YYYY	is the four-digit year
DDD	is the day-of-year (001-366)
:	is the colon character (0x3A)
HH	is the hour-of-the-day (00-23)
MM	is the minute-of-the-hour (00-59)
SS	is the second-of-the-minute (00-60)
d	is the DST indicator (S,I,D,O).
+ZZ	+ or - time zone offset relative to UTC (00-12)
<CR>	is the ASCII carriage return character (0x0D). The first <CR> is the on-time character.
<LF>	is the ASCII line feed character (0x0A).

NMEA Format

The National Marine Electronics Association (NMEA) has developed a specification that defines the interface between various pieces of marine electronic equipment. This standard defines "sentences" that contain GPS position, navigation, time, and other information. Sentences that have been added to the Unison product line are GGA, GLL, GSA, RMC, VTG and ZDA. However, position and navigation information is not available in a CDMA-Synchronized product so the only sentence that has been fully implemented is ZDA (time and date information).

ZDA (Time and Date)

The ZDA sentence identifies the time associated with the current 1PPS pulse. Each sentence is transmitted within 500 milliseconds after the 1PPS pulse is output and tells the time of the pulse that just occurred. If the Unison is unsynchronized then this sentence will be composed of null fields. Examples are below:

```
$GPZDA,,,,,,*48<CR><LF>
$GPZDA,175658.00,20,05,2008,07,00*69<CR><LF>
```

Msg ID	$GPZDA	"$" is the on-time character.
Field 1	175658.00	UTC time at 1PPS (hhmmss.ss)
Field 2	20	Day (01 to 31)
Field 3	05	Month (01 to 12)
Field 4	2008	Year (1980 to 2079)
Field 5	07	Local zone hour, offset from UTC (- for east longitude)
Field 6	00	Local zone minutes, offset from UTC
Checksum	*69	
Msg End	<CR><LF>	

Appendix *H*

Precision Time Protocol (PTP) IEEE-1588

The appendix contains the configuration and status information for the optional Precision Time Protocol. The PTP protocol running on the Unison is a full Grandmaster Clock implementation of the IEEE-1588 2002 standard.

About PTP

The PTP implementation in the Unison is built from the distribution at the PTPd website: http://ptpd.sourceforge.net. The PTP daemon **`ptpd`** status and configuration is supported from two PTP companion utilities **`cptpstat`** and **`ptpconfig.`** For more information about **`ptpd`** and to obtain PTP Slave software, refer to the PTPd website.

An excellent book which describes the PTP Master and Slave operation is:

Measurement, Control, and Communication using IEEE 1588,
John C. Eidson, Springer, November 2002.

More information on IEEE-1588 PTP can be found at the NIST National Institute of Standards and Technology IEEE 1588 website: http://ieee1588.nist.gov

Configuration and Status

The default PTP configuration settings in the Unison are shown below. If you need to modify these settings then you will need to reconfigure the PTP subsystem. You may perform the configuration from either a **`telnet`** or **`ssh`** session or the local RS-232 console. The default PTP settings are:

PTP Preferred:	TRUE
PTP Sync Interval:	Two seconds
PTP Subdomain Name:	DFLT
PTP Time Mode	UTC

Configuring PTP Using the Network Interface or Serial Port

The **`ptpconfig`** command starts an interactive shell script that will allow you to configure the PTP subsystem of the Unison. You will be prompted to set the PTP Preferred State: TRUE or FALSE; the PTP Sync Interval: 1,2,8,16, or 64 seconds; and the PTP Subdomain Name: DFLT, ALT1, ALT2, or ALT3 and the PTP Time Mode: PTP or UTC. Default PTP configuration settings are shown at the beginning of this Appendix.

One file is modified, /etc/ptp.conf. This is a non-volatile stored file in the FLASH disk /boot/etc directory. You must re-boot the Unison after running this script for the changes to take effect.

The following is a transcript of the question and answer configuration utility provided by ptpconfig. The user entered parameters are underlined:

Unison CDMA(root@cntp)-> ptpconfig

```
***************************************************************************
******************* Precision Time Protocol Configuration ****************
***************************************************************************
*
*    This script will allow you to configure the ptp.conf file
*    that controls the PTP daemon operation.
*
*    You will be able to configure the PTP sync_interval, preferred state,
*    subdomain_name, and time_mode.
*
*    The changes you make now will not take effect until you re-boot the
*    Unison . If you make a mistake, just re-run ptpconfig prior to
*    re-booting.
*
*    You will now be prompted for the necessary set up parameters.
*
***************************************************************************

---PTP preferred Configuration

Set PTP preferred (TRUE, FALSE) TRUE

---PTP sync_interval Configuration

Set the PTP sync_interval in seconds (1,2,8,16,64) 2

---PTP subdomain_name Configuration

Set the PTP subdomain_name (DFLT, ALT1, ALT2, ALT3) DFLT

---PTP time_mode configuration

Set the PTP time_mode (UTC or PTP) PTP

***************************************************************************
*
*    The Unison Precision Time Protocol IEEE-1588 configuration has
*    been updated.
*
*    Please re-boot now for the changes to take effect.
*
***************************************************************************
```

Now re-boot the system by issuing this command at the shell prompt:

```
shutdown -r  now
```

PTP Status Using the Network Interface or Serial Port

The **cptpstat** command allows the user to query the status of the PTP subsystem. The ptpd daemon running on the system updates the /var/log/ptp.monitor every five seconds under normal operation. This logfile is parsed and formated to provide the status string having these fields:

```
PTPMODE CKID Stratum SSS PPPPP SDOM II LL TMD V
```

Where:

PTPMODE	is the PTP port state, either Master or Passive.
CKID	is the sync identifier, either ATOM or GPS.
SSS	is the PTP stratum, either 3 or 255, where 3 implies an error of > 100 nS and 255 is the unsynchronized state.
PPPPP	is the PTP preferred master setting, either True or False.
SDOM	is the PTP subdomain, one of DFLT, ALT1, ALT2 or ALT3.
II	is the PTP sync update interval, one of 1,2,8,16, or 64, in seconds.
LL	is the PTP leap second setting, one of 59, 60, or 61, where 59 implies that the last minute of the current day will have 59 seconds (leap second deletion), and 61 implies that the last minute of the current day will have 61 seconds (leap second insertion).
TMD	is the PTP time epoch either PTP or UTC.
V	is the PTP protocol version, only version 1 is implemented.

Operation

The Unison is configured as an IEEE-1588 Grand Master Clock. Verify that the network settings have been configured and tested using **netconfig**. Once the network has been configured the Unison will begin to transmit PTP Sync messages to the slave clocks.

The PTP sync message and status report from the Unison is dependent on the status of the clock configuration including the oscillator type and CDMA receiver subsystem.

The port_state will report either MASTER or PASSIVE. MASTER is reported when the clock is locked to CDMA and the best master clock algorithm (BMC) designates this clock as the MASTER. The port_state will report PASSIVE if the clock has never locked to CDMA or if the BMC designates this clock as anything other than the MASTER.

The CDMA timing system defines the time to be GPS time. We call the CDMA timing technology “indirect GPS”. (For more information see ***Chapter 1 - CDMA Timing-How It Works***.) Therefore, the Unison will report the identifier as GPS when it is locked to CDMA.

The identifier will report either GPS, or ATOM. GPS is reported when the system starts and when locked to CDMA. ATOM is only reported when the oscillator type in the Unison is a Rubidium.

The PTP Stratum will report either 3 or 255. The identifier and the calculated offset to UTC determines the PTP Stratum as shown below:

Identifier	PTP Stratum	Offset to UTC
GPS or ATOM	3	> 100 nanosecs
GPS or ATOM	255	Never locked

The PTP leap_59 and leap_61 report either TRUE or FALSE. FALSE is reported when no leap insertion or deletion is pending. The leap_61 reports TRUE on the day of a leap second insertion. The leap_59 is TRUE on the day of a leap second deletion.

About the PTP Second and UTC Time

The PTP time_mode selections are PTP and UTC. The IEEE-1588 standard defines the PTP epoch beginning at 0 hours on 1 January 1970. The time measured since this epoch is designated in the standard as PTP seconds. The PTP second is monotonic and does not include leap seconds.

Unlike PTP, the UTC second is not monotonic, that is, from time-to-time there will be leap second insertions. The last second of a leap day is 23:59:60 making the day one second longer than a normal day ending at 23:59:59. See ***Appendix D - Leap Seconds*** for more information.

PTP Second

When the PTP time mode is set to PTP then the slave clocks must utilize the current leap second and leap second pending flags (leap_59 or leap_61) to convert the PTP second to UTC.

UTC Time

When the PTP time mode is set to UTC then there will be a one second jump in time when a leap second insertion occurs. If the PTP slave does not account for this, it will also jump. Avoid this by using PTP time mode.

PTP Software License

PTP as implemented in the Unison is cover by patents and copyrights.

See the IEEE Standards Association at: http://standards.ieee.org/db/patents/pat1390.html for patents that pertain the the Std No 1588.

Information about the PTP Project, led by Kendall Correll, can be found at ptpd.sourceforge.net The distribution and usage of the PTP software is allowed, as long as the following copyright notice is included in our documentation.

The following copyright notice applies to all files which compose the PTPd. This notice applies as if the text was explicitly included each file.

Appendix I

Specifications

CDMA Receiver:

Cellular Mobile Receive Band – 869-894 MHz (Standard)
North American PCS Mobile Receive Band – 1930-1990 MHz (Standard)
Japanese Cellular Mobile Receive Band – 832-870 MHz (Optional Configuration)
TIA/EIA IS-95 CDMA Pilot and Sync channels.

Antenna:

TNC jack on rear panel, $Z_{in} = 50\Omega$.
Dual Band, 824-896 MHz/1850-1990 MHz,
magnetic-base monopole with integral 12 ft. RG-58/U cable and TNC plug.
Extension cables and low noise pre-amplifiers are available as options.

Local Oscillator:

TCXO is standard ($2.5\text{x}10^{-6}$ over -20° to 70°C).
Option: Medium-Stability OCXO ($4\text{x}10^{-9}$ over 0 to 70°C).

Stratum 1 Holdover Performance:	24 Hours - TCXO
	35 Days - MS-OCXO

Time to Lock:

< 5 minutes, typical (TCXO).
< 10 minutes, typical (MS-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 CDMA acquisition and lock status.
Network LED: Amber LED indicates network activity.
Alarm LED: Red LED indicates a fault condition.

Maintenance Console:

RS-232 serial I/O on rear panel DB9M plug for secure, local terminal access.
Parameters fixed on 19200 baud, 8 data bits, no parity, 1 stop bit.
See ***RS-232 Serial I/O Port Signal Definitions*** in ***Chapter 5*** for more information.

Synchronization Accuracy:

CDMA Receiver Accuracy: <10 microseconds to UTC when locked, typical.
NTP Timestamp Accuracy: <10 microseconds @ 200 packets/second (200,000 clients).
NTP Client Synchronization Accuracy: Network factors can limit LAN synchronization accuracy to 1/2 to 2 milliseconds, typical.

Supported IPv4 Protocols:

SNTP, NTP v2, v3, v4 and broadcast/multicast mode; MD5 authentication and autokey
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 Protocols:

SNTP, NTP v2, v3, v4 and broadcast/multicast mode; MD5 authentication and autokey
SSH server with "secure copy" utility, SCP
SNMP v1, v2c, v3 with Enterprise MIB
TIME and DAYTIME server
SYSLOG

Optional PTP/IEEE-1588 Grandmaster:

IEEE-1588-2002 (V1)
PTP Timestamp Resolution: 1 microsecond.
PTP Slave Synchronization Accuracy to the Grandmaster: 10 microseconds, typical, network topology dependent.

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:	2" Dia. at base x 14" 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

Optional Outputs:

See ***Chapter 2 - Physical Description*** for more information on this output.

1PPS: Positive TTL pulse into 50Ω.
Width: User-selectable to 20 us, 1 ms, 100 ms, 500 ms.
Accuracy: <10 microseconds to UTC when locked.
Stability: TDEV < 50 ns, $\tau < 10^4$ seconds.
Connector: Rear-panel BNC labeled "1PPS".

AM Code: 1 Vrms into 50Ω, 1 kHz carrier.
Signal: Amplitude-modulated (AM), 3:1 ratio.
Format: User-selectable to IRIG-B (120/IEEE-1344, 122, 123), NASA-36, 2137.
Connector: Rear-panel BNC labeled "AM CODE".

Prog TTL Pulse Rate: Positive TTL pulse @ 50Ω on BNC jack.
Rate: User selectable to 1, 10, 100, 1K, 10K, 100K, 1M, 5M, 10M PPS and Timecode.
Duty Cycle: 50% except for 1PPS which mimics the 1PPS Output defined above.
Accuracy: $< 10^{-11}$ to UTC for 24-hour averaging times when locked.
Stability: $\sigma_y(\tau) < 10^{-9}$ for $\tau < 10^3$ seconds, $\sigma_y(\tau) < 10^{-6}/\tau$ for $\tau > 10^3$ seconds.
Connector: Rear-panel BNC jack labeled "PROG TTL".

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".

Serial Time: Output only port at RS-232 levels.
Baud Rate: User Selectable to 4800, 9600, 19200 or 57600.
Parity: User Selectable to Odd, Even or None.
ASCII Formats: User-Selectable to Sysplex, EndRun, EndRunX, Truetime, NENA, or NMEA.
Connector: Rear-panel DB-9M connector labeled "SERIAL TIME".
Pinout: Pin 3 is Transmit Data. Pin 5 is GND.
(See ***Appendix G - Serial Time Output*** for more information.)

1 PPS (RS-422): RS-422 Levels
Width: User selectable to 20 us, 1 ms, 100 ms, 500 ms.
Accuracy: < 10 microseconds to UTC when locked, typical.
Stability: TDEV < 50 ns, $\tau < 10^4$ seconds.
Connector: Rear-panel DB-9M jack labeled "1PPS RS-422".
Pinout: Pin 3 is +signal. Pin 6 is -signal. Pin 5 is GND.

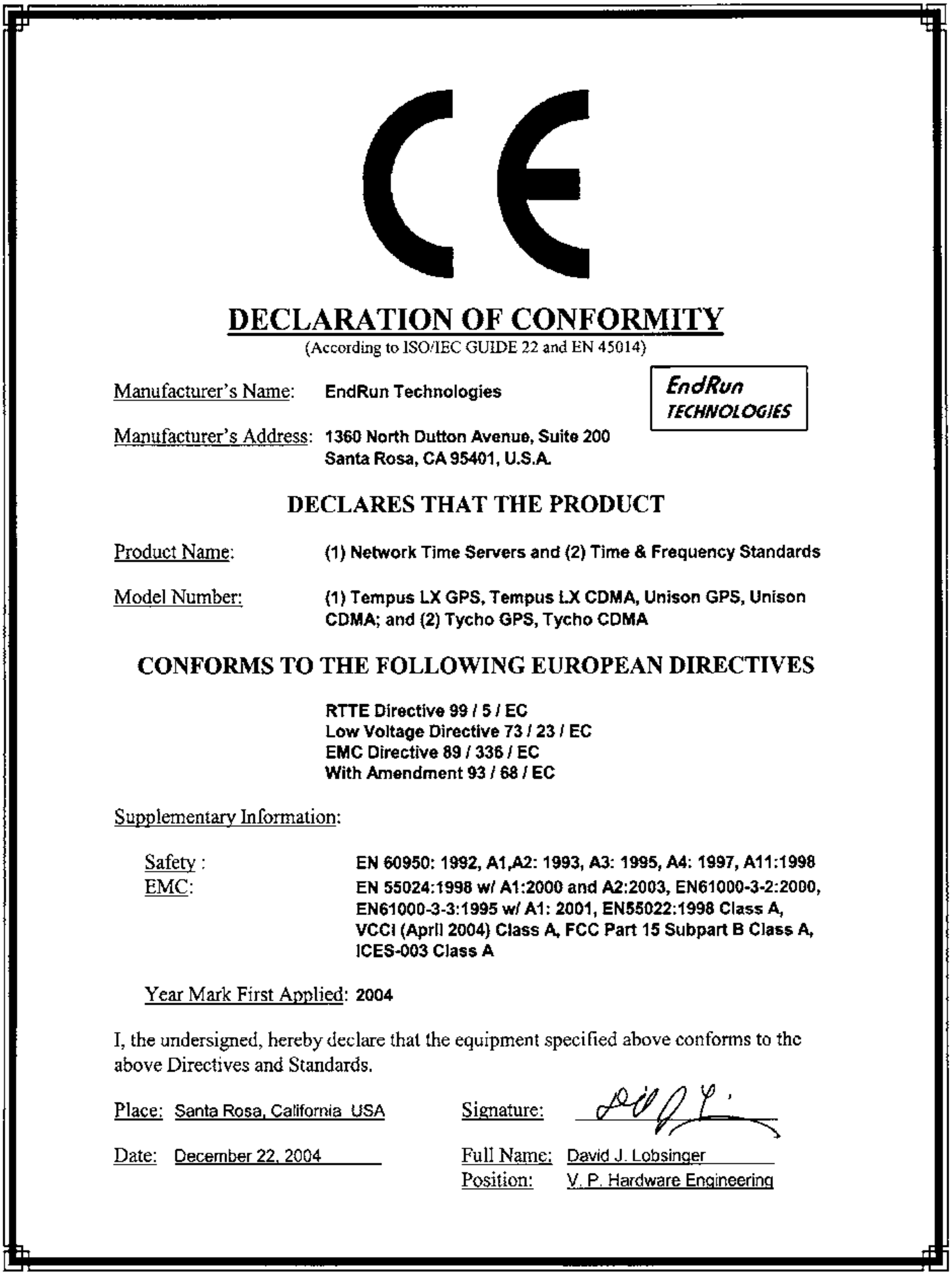
Fixed Rate: Positive TTL pulse @ 50Ω.
Rate: Preset at Factory and cannot be changed.
Accuracy: $< 10^{-11}$ to UTC for 24-hour averaging times when locked.
Stability: $\sigma_y(\tau) < 10^{-9}$ for $\tau < 10^3$ seconds, $\sigma_y(\tau) < 10^{-6}/\tau$ for $\tau > 10^3$ seconds.
Connector: Rear-panel BNC jack labeled with appropriate rate such as "10MPPS".

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)

EndRun TECHNOLOGIES

Manufacturer's Name: 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

Special Modifications

Changes for Customer Requirements

From time to time EndRun Technologies will customize the standard Unison Network Time Server for special customer requirements. If your unit has been modified then this section will describe what those changes are.

This section is blank.

EndRun
TECHNOLOGIES
"Smarter Timing Solutions"

2270 Northpoint Parkway
Santa Rosa, CA 95407
TEL 1-877-749-3878
FAX 707-573-8619
www.endruntechnologies.com