EndRun TECHNOLOGIES
1360 North Dutton Avenue #200, Santa Rosa, CA 95401

Version 2.60 Firmware Upgrade Release Notes
(Meridian, Tempus LX and Unison Product Families)

Version 2.60 of the Linux root file system (RFS) and the new IPv6-capable Linux 2.4.31-IPV6 kernel are now shipping in new products. Both files are also available for download so that you can upgrade your installed units in the field. This is a major upgrade, and features updated versions of all applications, utilities and shared libraries typically installed in an embedded Linux-based system. In addition, the critical open source protocol implementations, NTP, OpenSSH, Net-SNMP and Syslog-ng are now IPv6-capable, along with other various support daemons and configuration utilities that need to understand IPv6 addresses. The NTP implementation is now capable of "autokey" cryptographic operation. Previously, only symmetric MD5 cryptography was available.

Easy Field Installable Upgrade
The new system detects the presence of an IPv6-capable kernel and enables the IPv6 configuration menus and command line utilities automatically. As with all of our firmware upgrades, we have designed the upgrade to be as seamless as possible for existing customers, which means that after applying the update, your existing configuration settings and passwords will continue to function properly. However, due to the magnitude of the changes included in this upgrade, there are a couple of cases where configuration files must be re-configured:

If You Are Using DHCP
The new version DHCP client daemon included in the 2.60 RFS will by default overwrite the */etc/ntp.conf*. This will cause serious problems. If you have a pre-existing */boot/etc/rc.d/rc.inet1* that is set up to invoke **`dhcpcd`** to configure the ethernet interface, you will need to re-run **`netconfig`** immediately after performing the upgrade and re-boot. This will replace the old */boot/etc/rc.d/rc.inet1* with a new one that will invoke **`dhcpcd`** with the appropriate arguments to inhibit this behavior.

If You Are Operating NTP Without MD5 Authentication
The new version NTP server daemon included in the 2.60 RFS interprets certain keywords in the "restrict" directive differently than the previous version. In particular, it will now interpret the "notrust" keyword to mean that it will not reply to client requests that do not use authentication (MD5 or autokey). Previous versions of the NTP server did not operate this way. If you have a pre-existing */boot/etc/ntp.conf,* and any of your NTP clients are configured to not use MD5 authentication, you should re-run **`ntpconfig`** immediately after performing the upgrade and re-boot. This will replace the old */boot/etc/ntp.conf* with a new one that will have the "notrust" keyword removed from the "restrict" directive. The new file will also contain the "keysdir" directive to support operation with autokey.

EndRun TECHNOLOGIES
1360 North Dutton Avenue #200, Santa Rosa, CA 95401

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

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

```
upgradekernel /tmp/newkernelimage
```

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

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

OpenSSH
Starting with the 2.60 RFS, **`sshd`** is no longer started by the superserver daemon, **`inetd`**. If you have a previously reconfigured */boot/etc/inetd.conf*, the */etc/rc.d/rc.inet2* startup script will detect it and remove the line that allows **`sshd`** to be started by **`inetd`**. By default, **`sshd`** is factory configured to

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

Net-SNMP

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

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

New IPv6-Capable syslog-ng

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

Remaining IPv4-Only Protocols

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